Wang *et al. EURASIP Journal on Wireless Communications and Networking* (2019) 2019:12
https://doi.org/10.1186/s13638-018-1327-7

EURASIP Journal on Wireless Communications and Networking

RESEARCH Open Access

An efficient construction of polar codes based on the general partial order

Wei Wang[1], Liping Li[1*] and Kai Niu[2]

Abstract

Traditionally, the construction of polar codes requires intense computations to sort all bit channels. In previous works, two types of partial orders (POs) of polar codes were proposed to decrease the computations in the construction process. In this paper, a procedure is presented to employ POs with complexity $O(N^2)$, which is lower than the existing procedures, where $N = 2^n$ is the code length ($n \geq 1$). To further reduce the computation complexity, in this paper, we propose a general partial order (GPO), which works at a lower dimension ($n_u < n$) to sort bit channels. The principle of the GPO is to divide the binary expansion of bit channels into two parts: the upper part and the lower part. First, the relationships among bit channels formed from the upper part (with length n_u) are completely determined. Existing algorithms are called in this new level n_u when needed. Second, we prove that the ordering at this lower dimension n_u can be combined with the PO ordering of bit channels formed from the lower part. Therefore, more intrinsic relationships among bit channels are obtained. Working at the lower dimension $n_u < n$, the complexity of GPO is inherently smaller than the existing sorting algorithms. The studies show that for $n = 10$ and the code rate $R = 0.5$ (the worst code rate from the empirical studies), the GPO can order 82% of bit channels, which is 32% larger than that obtained from using only POs. The proposed GPO in this paper can significantly reduce computations in the construction of polar codes.

Keywords: Polar code, Polar code construction, Partial orders

1 Introduction

Polar codes [1] have attracted researchers from around the world since their introduction in [1]. As proven in [1], polar codes can achieve the channel capacity at a low encoding and decoding complexity of $\mathcal{O}(N\log N)$ for binary-input, memoryless, output-symmetric (BMS) channels (here, N is the code length). A successive cancelation (SC) decoder was proposed to decode N synthesized bit channels $W_N^{(i)}$, $i = 1, 2, \cdots, N$. The theoretical analysis of the polarization rate of bit channels is reported in [1–4]. The performance of polar codes with practical code lengths is investigated in [5–8] and their references.

The construction problem of polar codes is as follows: for the given code rate R, how to select the $K = \lfloor NR \rfloor$ best bit channels from all bit channels $\left\{W_N^{(i)}\right\}_{i=1}^{N}$ to transmit information bits. The inputs of the remaining bit channels are frozen and are made known to the transmitter and the receiver. The quality of a bit channel $W_N^{(i)}$ can be measured, for example, by the error probability $P_e\left(W_N^{(i)}\right)$. However, determining the quality of the bit channels is not an easy task with the exception of binary erasure channels (BECs). Note that the channel input alphabet size is assumed to be two. Non-binary input alphabets in [9–11] are not discussed in this paper.

Let $\mathcal{I}$ denote the set containing K indices of the good bit channels, and let $\mathcal{F}$ denote the set containing the indices of the frozen bit channels. Let $W : \mathcal{X} \to \mathcal{Y}$ be the underlying channel with a transition probability $W(y|x)$. Sorting the bit channels is difficult because of the large number of output alphabets $|\mathcal{Y}|^N$. Evaluating this large number of transition probabilities for each bit channel is beyond the reach of a practical system. Monte-Carlo simulations are proposed in [1] to sort the bit channels, with a high complexity of $\mathcal{O}(MN\log N)$

*Correspondence: liping_li@ahu.edu.cn
[1] Key Laboratory of Intelligent Computing and Signal Processing of the Ministry of Education of China, Anhui University, Hefei, People's Republic of China
Full list of author information is available at the end of the article

(M denotes the iterations of the Monte-Carlo simulations). Density evolution is used in [12] to construct polar codes. The density evolution process involves function convolutions whose numerical calculation precisions are limited by the complexity. In [13], bit channel approximations are proposed with a controlled complexity of $\mathcal{O}\left(N \cdot \mu^2 \log \mu\right)$ (μ represents a user-defined parameter to limit the number of output alphabet at each approximation stage). Another family of polar code construction is to use Gaussian approximation (GA) on additive white Gaussian noise (AWGN) channels [14–16]. The GA method is inherently limited by the approximation function [16] and may not be applicable to all block lengths. Partial order (PO) relations are reported in [12, 17, 18]. We borrow the notations from [17] to denote the two types of POs: $\preceq_1$ and $\preceq_2$. The first type of PO, $\preceq_1$, is investigated in [17] and [18]; it orders the bit channels with the same Hamming weight (here, the Hamming weight is the Hamming weight of the binary expansions of the bit channel indices). The second type of PO, $\preceq_2$, is stated in [12], which orders the channels with different Hamming weights. These two POs are channel independent: the orders are universal for all types of underlying channels. In [17], the authors show that the two POs can be used to simplify the encoding construction. However, no algorithm is presented to employ these two POs; therefore, no detailed results are provided there in [17] to quantify the exact savings.

In this paper, an efficient procedure is proposed to employ the two POs $\preceq_1$ and $\preceq_2$. Theoretical foundation enabling this efficient procedure is provided in Proposition 1 and Lemma 1–Lemma 3. Compared with the work in [19], the proposed procedure to find the relationships determined by the two POs has a lower complexity: our proposed algorithm has a complexity of $\mathcal{O}\left(N^2\right)$ while the complexity in employing the two POs is at least $\mathcal{O}\left(N^{5/2}\right)$ in [19]. Although both procedures can be performed offline, the proposed procedure in this paper is more efficient in employing POs. Note that the proposed procedure in the paper can be refined by exploring the nesting and symmetric properties of POs in [20].

For a given channel W, to reduce the computation complexity of the construction of polar codes, we introduce a general partial order (GPO), which works by dividing the binary expansion of bit channels into two parts: the upper part with $n_u \geq 1$ bits and the lower part with $n_l = n - n_u$ bits (here, $n = \log_2 N$). Considering the upper part as a set of new bit channels with n_u bits, the ordering of these new bit channels at this smaller level $N_u = 2^{n_u}$ can either be derived from the two POs or from calling any sorting algorithm [12, 13]. Since the channel independent ordering of N_u bit channels is part of POs, we only consider their new channel-dependent ordering obtained from the existing sorting procedures such as [12, 13]. This paper proves that this channel-dependent ordering of N_u bit channels can be combined with $\preceq_1$ and $\preceq_2$ to obtain the ordering for the given underlying channel W and block length N. The POs and GPO aim to find the intrinsic relationship between any two bit channels. After applying POs and GPO, additional intrinsic relationships can be identified when considering the transitivity of bit channel degradation (or upgradation). These transitive relations are termed indirect relations. Algorithms to explore PO $\preceq_1$, PO $\preceq_2$, GPO, and the transitivity of bit channel degradation (or upgradation) are provided to quantify the savings of each. The choice of the dimension n_u is also discussed in the paper. A quasi-analytical equation that can be used to estimate the optimal n_u, which produces the most efficient construction of polar codes, is provided. The calculations in our study indicate that for $n = 10$ and the code rate $R = 0.5$ (the worst code rate), incorporating PO relations can determine 50% of the bit channels, which produces only 50% of the sorting calculations. With our proposed GPO, the determined portion of the channels increases to 82%, which causes a significant complexity reduction in the construction of polar codes.

Note that portions of this work are presented in [21], where three lemmas of the procedure to find relationships determined by POs are not introduced. The theorem of GPO is not proven and the indirect relationships are not considered due to the limited space of [21]. The main contributions of this current paper are as follows: (1) The three lemmas related to POs are introduced. These lemmas are the basis for the proposed procedure in finding relationships determined by PO $\preceq_1$ and PO $\preceq_2$. The complexity of the proposed procedure is lower than the existing procedure when employing the two POs. (2) A complete proof of the proposition related to GPO is given. The procedures of GPO are channel-independent while the ordering obtained from GPO is channel dependent. (3) A proposition to find all indirect relationships is presented and proven. (4) The GPO parameter is optimized. The simulated construction time follows the prediction of the construction time and shows a construction time around 30% of the original Tal-Vardy's construction time for code lengths up to $N = 1024$. (5) The algorithms for implementing POs and GPO are refined with specific mathematical expressions due to new introduced mathematical details compared with [21].

The remainder of the paper is organized as follows. Section 2 introduces the basics of polar codes and the partial orders we employed in this paper. The GPO is proposed in Section 3, with algorithms and proofs presented in this section. The discussion of the selection of the low dimension n_u is also provided in Section 3. Section 4 provides the numerical results that quantify the savings of the proposed construction algorithms. Conclusion remarks are presented at the end in Section 5.

2 Polar code and partial orders

The first part of this section includes the relevant basics of polar codes from [1]. For other details of polar codes, please refer to the materials in [1]. The second part restates some results of partial orders from [12, 17].

2.1 Notation specification

The notations in this paper are as follows: The notation v_1^N is used to represent a row vector with elements $(v_1, v_2, ..., v_N)$. The n-bit binary expansion of the integer i is written as $i = (i_n, i_{n-1}, ..., i_1)_b$. Given the vector v_1^N, the vector v_i^j is a subvector $(v_i, ..., v_j)$ with $1 \leq i,j \leq N$. For a set $\mathcal{A} \in \{1, 2, ..., N\}$, $v_{\mathcal{A}}$ denotes a subvector with elements in $\{v_i, i \in \mathcal{A}\}$.

2.2 Polar codes

Let W be any BMS channel with the transition probability $W(y|x)$. The input alphabet $\mathcal{X}$ takes values in $\{0, 1\}$, and the output alphabet is $\mathcal{Y}$. The generator matrix is $G = BF^{\otimes n}$, where B is a bit-reversal permutation matrix and $F = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}$. The operation $F^{\otimes n}$ is the nth Kronecker power of F over the binary field $\mathbb{F}_2$. Polar codes synthesize $N = 2^n (n \geq 1)$ bit channels, $\left\{ W_N^{(i)} \right\}_{i=1}^N$, out of N independent copies of W. The transformation has a tree structure in [1], which has a basic one-step channel transformation defined as $(W, W) \mapsto \left(W^{'}, W^{''} \right)$, where

$$W^{'} (y_1, y_2|u_1) = \sum_{u_2} \frac{1}{2} W (y_1|u_1 \oplus u_2)\, W (y_2|u_2) \tag{1}$$

$$W^{''} (y_1, y_2, u_1|u_2) = \frac{1}{2} W (y_1|u_1 \oplus u_2)\, W (y_2|u_2) \tag{2}$$

A bit channel i with $i - 1 = (i_n, i_{n-1}, ..., i_1)_b$ $(1 \leq i \leq N$, the left most bit is the MSB) is transformed in each tree level according to the bit of $i - 1$ at that level: at tree level $1 \leq k \leq n$, bit channel i takes $W^{'}$ if $i_k = 0$. Otherwise, $W^{''}$ is taken at level k. The Bhattacharyya parameters of $W^{'}$ and $W^{''}$ satisfy the following conditions:

$$Z\left(W^{''}\right) = Z(W)^2 \tag{3}$$

$$Z\left(W^{'}\right) \leq 2Z(W) - Z(W)^2 \tag{4}$$

$$Z\left(W^{'}\right) \geq Z(W) \geq Z\left(W^{''}\right) \tag{5}$$

Note that for binary erasure channels (BEC), the Bhattacharyya parameter $Z\left(W^{'}\right)$ has an exact expression $Z\left(W^{'}\right) = 2Z(W) - Z(W)^2$, resulting in a recursive calculation of the Bhattacharyya parameters of the final bit channels. Finally, after the channel transformations, the transition probability for bit channel i is defined as

$$W_N^{(i)} \left(y_1^N, u_1^{i-1}|u_i \right) = \sum_{u_{i+1}^N \in \mathcal{X}^{N-i}} \frac{1}{2^{N-1}} W^N \left(y_1^N|u_1^N G \right) \tag{6}$$

where $W^N(\cdot)$ is the underlying vector channel (N copies of the channel W).

2.3 Channel degradation

First, we restate the concept of stochastic channel degradation as presented in [13]. For a channel $Q : \mathcal{X} \rightarrow \mathcal{Z}$ and channel $W : \mathcal{X} \rightarrow \mathcal{Y}$, Q is stochastically degraded with respect to W if there exists an intermediate channel $P : \mathcal{Y} \rightarrow \mathcal{Z}$ such that

$$Q(z|x) = \sum_{y \in \mathcal{Y}} W(y|x) P(z|y) \tag{7}$$

for all $x \in \mathcal{X}$ and $z \in \mathcal{Z}$. In this case, we write $Q \preceq W$ (or $W \succeq Q$) as in [13].

As proven in both ([22], p. 207) and [13], for BMS channels, the probability of error and the Bhattacharyya parameter preserves degradation. Let $P_e(W)$ denote the error probability when transmitting a single bit over the channel W. Let Q and W be two BMS channels and Q is stochastically degraded with respect to W: $Q \preceq W$. Then, from ([22], p. 207) and [13], the following statements hold:

$$P_e(Q) \geq P_e(W) \tag{8}$$

$$Z(Q) \geq Z(W) \tag{9}$$

Note that in [1], Arıkan suggested choosing bit channels with the smallest Bhattacharyya parameters as the good channel indices, that is, if the set $\mathcal{I}$ contains the good bit channels, then for any $i \in \mathcal{I}$ and any $j \notin \mathcal{I}$: $Z\left(W_N^{(i)}\right) \leq Z\left(W_N^{(j)}\right)$. In [13], the error probabilities of all approximated bit channels are calculated. The information set $\mathcal{I}$ therein contains all the bit channels with the smallest error probabilities. In summary, [1, 13] utilize (9) and (8), respectively, as criterions to select the good bit channels.

2.4 Partial orders

Consider two bit channels i and j with binary expansions $i - 1 = (i_n, i_{n-1}, ..., 1)_b$ and $j - 1 = (j_n, j_{n-1}, ..., 1)_b$. In [17], if the binary expansion of $j - 1$ can be obtained from the binary expansion of $i - 1$ by switching some higher position 1s with some lower position 0s, then $W_N^{(j)}$ is stochastically degraded with respect to $W_N^{(i)}$. This order is written as $\preceq_1$, which is equivalent to $\preceq$. The PO $\preceq_1$ is transitive. Here, is an example of this PO $\preceq_1$.

Let $N = 2^4 = 16$. Consider bit channels $i = 13$ and $j = 11$ with binary expansions $i - 1 = 13 - 1 = (1100)_b$ and $j - 1 = 11 - 1 = (1010)_b$. From the binary expansion of $i - 1$, it can be seen that switching value 1 at position 3 with value 0 at position 2 results in the binary expansion of $j - 1$. Therefore, bit channel j is degraded with respect to bit channel i, irrespective of the underlying channels.

The second PO is from [12], which is denoted as $\preceq_2$. For the two bit channels i and j, if every bit of the binary expansion of $i-1$ is greater than or equal to every bit of the binary expansion of $j - 1$, then $W_N^{(j)} \preceq_2 W_N^{(i)}$. With a symmetric underlying channel W, $W_N^{(j)} \preceq_2 W_N^{(i)} \implies W_N^{(j)} \preceq W_N^{(i)}$ is proven in [12]. Therefore, the second partial order $\preceq_2$ can also be employed to order the bit channels. The PO $\preceq_2$ is also transitive. An example of this PO $\preceq_2$ also adopts $N = 2^4 = 16$. Consider bit channels $k = 15$ and $i = 13$, with binary expansions of $k - 1 = 15 - 1 = (1110)_b$ and $i - 1 = 13 - 1 = (1100)_b$. It can be seen that for these two bit channels, the two ones of the binary expansion of $i - 1$ are the same as those of $k - 1$ at the same positions. However, at position 2, the value of the binary expansion of $k - 1$ (value 1) is greater than that of $i - 1$ (value 0). This is exactly PO $\preceq_2$. Therefore, bit channel i is degraded with respect to bit channel k. Combining the example in PO $\preceq_1$, it can be concluded that $W_{16}^{(11)} \preceq W_{16}^{(13)} \preceq W_{16}^{(15)}$, irrespective of the underlying channels.

The constructions in [12–16] sort all N bit channels. The inherent ordering of polar codes from $\preceq_1$ and $\preceq_2$ is not utilized before sorting all bit channels. Therefore, the amount that we can save in the construction by applying POs is not clear. In this paper, we apply $\preceq_1$ and $\preceq_2$ before sorting all N bit channels and to quantify the savings in this process. Three sets, which correspond to the good bit channels ($\mathcal{I}$), the frozen bit channels ($\mathcal{F}$), and the undetermined bit channels ($\mathcal{U}$), are selected by applying the two POs. If the number of bit channels in $\mathcal{I}$ is smaller than K, then a sorting algorithm, for example, the approximation in [13], can be applied to sort the bit channels in $\mathcal{U}$ to select $K - |\mathcal{I}|$ best bit channels from $\mathcal{U}$ and place those $K - |\mathcal{I}|$ best bit channels in $\mathcal{I}$. (Note that by fixing the number of information bits K in one code block, it is always that $|I| \leq K$ since the bit channels are compared among themselves, and at most the best K can be selected.) The smaller is the size of $\mathcal{U}$, the larger are the savings. The POs are channel independent and universal for all BMS channels. Therefore, the sets $\mathcal{I}$, $\mathcal{U}$, and $\mathcal{F}$ for the given block length N and the code rate R need to be calculated only once and can be calculated off-line. Note that the final K best bit channels that are selected from applying the two POs before calling any sorting algorithm from [12–16] should be the same as calling the sorting algorithm since the good bit channels selected from the POs are channel independent.

3 Construction combining POs and GPO

In this section, the definition of the partial order indicator (POI) and the implementation of the POs is first introduced. The detailed theory and a proof of the general partial order (GPO) are presented in Section 3.2. Then, indirect relationships after applying POs and GPO are discussed in Section 3.3. The optimal choice of the parameter $n_u < n$, on which the GPO works, is discussed in 3.4. At last, a top-level algorithm of the proposed general partial order (GPO) is provided.

Given the three sets defined in Section 2.4, the following proposition can be used to determine whether a bit channel should be in $\mathcal{I}$, $\mathcal{F}$, or $\mathcal{U}$.

Proposition 1 *If a bit channel i is stochastically upgraded with respect to at least $N - K$ bit channels, then $i \in \mathcal{I}$. If a bit channel j is stochastically degraded with respect to at least K bit channels, then $j \in \mathcal{F}$. All other bit channels are in the undetermined set $\mathcal{U}$.*

Because the proof of this proposition is straightforward, it is omitted here.

3.1 Definition of POI matrix and implementation of POs

The matrix O is defined as an $N \times N$ matrix whose element (i, j) is the partial order indicator (POI) between bit channel i and bit channel j $(i > j)$. The matrix O is a lower triangular matrix (excluding the diagonal), and the POIs of O can be written as:

$$O_{ij} = \begin{cases} 1, & \text{if } i > j \text{ and } W_N^{(i)} \succ W_N^{(j)} \\ -1, & \text{if } i > j \text{ and } W_N^{(i)} \prec W_N^{(j)} \\ 0, & \text{if } i > j \text{ but } W_N^{(i)} \text{ and } W_N^{(j)} \\ & \quad\text{is unknown} \\ 0, & \text{if } i \leq j \end{cases} \tag{10}$$

In practical implementations, the unused part of O (the upper triangular part) can be saved. Algorithm 1 finds the POIs when applying PO $\preceq_1$ and PO $\preceq_2$.

To explain Algorithm 1, the following lemma is introduced first.

Lemma 1 *For a polar code with the code length N, consider two bit channels $1 \leq i, j \leq N$ with $j < i$. Then, purely from PO $\preceq_1$ and PO $\preceq_2$, it is impossible for bit channel j to be stochastically upgraded with respect to bit channel i.*

Proof The proof is straightforward. The binary expansions of the two bit channels are $i - 1 = (i_n, i_{n-1}, ..., i_1)_b$ and $j - 1 = (j_n, j_{n-1}, ..., j_1)_b$. With $i > j$, there must be a position k $(1 \leq k \leq n)$ at which (1) $i_k = 1$ but $j_k = 0$; (2) for positions larger than k, $(i_n, i_{n-1}, ..., i_{k+1}) = (j_n, j_{n-1}, ..., j_{k+1})$. There is no requirement for positions smaller than k. These two conditions are the basic requirements for $i > j$. With these two conditions, PO $\preceq_1$ cannot be applied to $j-1 = (j_n, j_{n-1}, ..., j_1)_b$ to reach any intermediate channel v equal or upgraded with respect to bit channel i: with $(i_n, i_{n-1}, ..., i_{k+1}) = (j_n, j_{n-1}, ..., j_{k+1})$, there is not a larger position $k' > k$ so that $i_{k'} = 1$ and $j_{k'} = 0$ to switch with position k in order to make $j_k = 1$. With $i_k = 1$ and $j_k = 0$ and PO $\preceq_1$ not applicable, PO $\preceq_2$ naturally is not valid: $j_k < i_k$. Therefore, for $j < i$, with PO $\preceq_1$ and PO $\preceq_2$, it is impossible for bit channel j to be stochastically upgraded with respect to bit channel i. □

Remark 1 *Lemma 1 does not mean that bit channel j cannot be stochastically upgraded with respect to bit channel i when j < i. Lemma 1 has a condition: from the POs point of view, bit channel j cannot be stochastically upgraded with respect to bit channel i, indicating that* $O_{i,j} \neq -1$ *according to the definition of (10). POs are channel independent. When it comes to any specific channel, the relationship between bit channel j and bit channel i with j < i can be evaluated using an existing sorting procedure. For example, with* $N = 16$*, consider the two bit channels* $i - 1 = 10 - 1 = (1001)_b$ *and* $j - 1 = 8 - 1 = (0111)_b$*. From the two POs, bit channel i is not stochastically upgraded with respect to bit channel j and bit channel j cannot be stochastically upgraded to bit channel i either from Lemma 1. However, consider an AWGN channel with a SNR of 1 dB, bit channel j is found to be stochastically upgraded with respect to bit channel i.*

For any two bit channels $i-1 = (i_n, i_{n-1}, ..., i_1)_b$ and $j-1 = (j_n, j_{n-1}, ..., j_1)_b$, denote the following two sets:

$$\mathcal{P} = \{1 \leq k \leq n \mid i_k = 1 \text{ and } j_k = 0\} \tag{11}$$

$$\mathcal{M} = \{1 \leq k \leq n \mid i_k = 0 \text{ and } j_k = 1\} \tag{12}$$

Let $p = |\mathcal{P}|$ and $m = |\mathcal{M}|$. At all the other positions except those specified by $\mathcal{P}$ and $\mathcal{M}$, the binary expansions of $i-1$ and $j-1$ share the same values zeros or ones. Assume the sets $\mathcal{P}$ and $\mathcal{M}$ are in descending order.

The following two lemmas state the conditions when PO $\preceq_1$ and PO $\preceq_2$ can (or cannot) be applied.

Lemma 2 *For a polar code with the block length* $N = 2^n$*, consider two bit channels* $i-1 = (i_n, i_{n-1}, ..., i_1)_b$ *and* $j-1 = (j_n, j_{n-1}, ..., j_1)_b$ *with* $i > j$*. The sets* $\mathcal{P}$ *and* $\mathcal{M}$ *are defined in (11) and (12), respectively. When* $m = |\mathcal{M}| = 0$*, then from PO* $\preceq_2$*, bit channel j is stochastically degraded with respect to bit channel i.*

Proof When $m = |\mathcal{M}| = 0$, it means there is no position k $(1 \leq k \leq n)$ at which $j_k = 1$ and $i_k = 0$. In other words, all values of the binary expansion of $j-1 = (j_n, j_{n-1}, ..., j_1)_b$ are equal or smaller than that of $i-1$. With $i > j$, invoking the definition of PO $\preceq_2$, it can be concluded that bit channel j is stochastically degraded with respect to bit channel i. □

Before introducing the following lemma, let us define that $\mathcal{P}_1^m$ contains element one to element m of the set $\mathcal{P}$. Also, the expression $\mathcal{P}_1^m > \mathcal{M}_1^m$ indicates the elementwise comparison between $\mathcal{P}_1^m$ and $\mathcal{M}_1^m$.

Lemma 3 *For a polar code with the block length* $N = 2^n$*, consider two bit channels* $i-1 = (i_n, i_{n-1}, ..., i_1)_b$ *and* $j-1 = (j_n, j_{n-1}, ..., j_1)_b$ *with* $i > j$*. The sets* $\mathcal{P}$ *and* $\mathcal{M}$ *are defined in (11) and (12), respectively, and are in descending order. If* $p = |\mathcal{P}| \geq m = |\mathcal{M}|$ $(m > 0)$ *and* $\mathcal{P}_1^m > \mathcal{M}_1^m$*, then PO* $\preceq_1$ *can be applied to achieve that bit channel j is stochastically degraded with respect to bit channel i.*

Proof It can be first observed that when $p < m$, PO $\preceq_1$ cannot be applied to $i-1 = (i_n, i_{n-1}, ..., i_1)_b$ to obtain any intermediate channel equal or stochastically upgraded with respect to $j-1 = (j_n, j_{n-1}, ..., j_1)_b$: there are not enough positions with values ones in $i-1 = (i_n, i_{n-1}, ..., i_1)_b$ (denoted by the set $\mathcal{P}$) to switch to positions in $\mathcal{M}$ because $p < m$. Therefore, to apply PO $\preceq_1$, $p \geq m$ is necessary. Consider the definition of PO $\preceq_1$: for a bit channel i, switching a higher position with value one with a lower position with value zero resulting in a channel which is degraded with respect to bit channel i. Therefore, with the descending sets $\mathcal{P}$ and $\mathcal{M}$, when $\mathcal{P}_1^m > \mathcal{M}_1^m$, PO $\preceq_1$ can be performed to $i-1 = (i_n, i_{n-1}, ..., i_1)_b$: for each element in $\mathcal{M}$, there is a larger element in $\mathcal{P}$ to switch with. Bit channel j is therefore stochastically degraded with respect to bit channel i. □

Remark 2 *Lemma 3 is actually a combination of PO* $\preceq_1$ *and PO* $\preceq_2$*. There is an inherent PO* $\preceq_2$ *in Lemma 3 as explained below. Let the intermediate channel after switching* $\mathcal{P}_1^m$ *with* $\mathcal{M}_1^m$ *of bit channel i as bit channel v. Then, bit channel j is stochastically degraded with respect to bit channel v. If* $p = m$*, then* $v = j$*, which is the case of PO* $\succeq_1$*. If* $p > m$*, there are remaining elements in* $\mathcal{P}$ *after switching with elements in* $\mathcal{M}$*. Comparing bit channel j and bit channel v, the new set* $\mathcal{M} = \emptyset$ $(m = 0)$*. From Lemma 2, bit channel v is stochastically upgraded with respect to bit channel j by invoking PO* $\preceq_2$*. But for our purpose of determining the relationship between bit channel i and bit channel j, the proof of Lemma 3 is sufficient.*

Algorithm 1 implements Lemmas 1 to 3. For any given bit channel i, bit channel $j < i$ is compared with it (line 6 in Algorithm 1). In Algorithm 1, −1s do not exist in O seen from Lemma 1. After running Algorithm 1, bit channels can be assigned to sets $\mathcal{I}$, $\mathcal{F}$, and $\mathcal{U}$ according to Proposition 1. As discussed in Section 2.4, the smaller is the size of $\mathcal{U}$, the smaller the number of remaining computations is.

Algorithm 1 Find all the POIs that POs can determine, and store the results in the matrix O

```
Require:
    $n$;
Ensure:
    a matrix $O$;
$N = 2^n$
$O \leftarrow (0)_{N \times N}$ // establish an $N \times N$ matrix $O$ and initialize the value in matrix $O$ as 0
for $i \leftarrow 2$ to $N$ do
    $a = (i_n, i_{n-1}, ..., i_1)_b$
    // From Lemma 1, only bit channel $j < i$ is compared with $i$
    for $j \leftarrow 1$ to $i - 1$ do
        $b = (j_n, j_{n-1}, ..., j_1)_b$
        $x = a - b$
        positions of $(x == 1) \rightarrow \mathcal{P}$
        positions of $(x == -1) \rightarrow \mathcal{M}$
        $m = |\mathcal{M}|$ // the length of $\mathcal{M}$
        if $m == 0$ then
            $O_{ij} = 1$ // PO $\preceq_2$ from Lemma 2
        else
            // POs $\preceq_1$ and $\preceq_2$ from Lemma 3
            if $|\mathcal{P}| \geq m$ then
                // element-wise comparison of descending sets
                if $P(1:m) > M(1:m)$ then
                    $O_{ij} = 1$
                end if
            end if
        end if
    end for
end for
return $O$;
```

3.2 General partial order (GPO)

The relationship between some bit channels cannot be determined from POs $\preceq_1$, $\preceq_2$, or their combinations. Here is an example: For $N = 2^8$, consider bit channel $i - 1 = (10011110)_b$ and $j - 1 = (01101011)_b$. Applying the two POs and their combinations yields no decision regarding whether $W_N^{(i)}$ is stochastically upgraded or degraded with respect to $W_N^{(j)}$. For general BMS channels, the relationship between bit channel i and bit channel j cannot be determined: $W_N^{(i)} \preceq W_N^{(j)}$ holds for one given underlying channel W; however, this relationship may not hold for another underlying channel W'. For a specific channel BMS channel W, we can further reduce the size of $\mathcal{U}$. Let us consider the two bit channels in the previous example. Let $\mathcal{Z}_u$ denote the upper part of the descending set $\mathcal{Z}_n = \{n, n-1, \ldots, 1\}$ with $n_u = |\mathcal{Z}_u|$. Then, $\mathcal{Z}_u$ is $\mathcal{Z}_u = \{n, n-1, \ldots, n-n_u+1\}$. Similarly, $\mathcal{Z}_l$ represents the lower parts of $\mathcal{Z}_n$ with $n_l = |\mathcal{Z}_l|$, and $\mathcal{Z}_l = \{n_l, n_l - 1, \ldots, 1\}$. The sizes satisfy $0 \leq n_u, n_l \leq n$ and $n_u + n_l = n$. Divide the binary expansion of $i - 1$ into two consecutive parts: $i - 1 = (i_{\mathcal{Z}_u}, i_{\mathcal{Z}_l})_b$. The upper part $i_{\mathcal{Z}_u}$ defines a new bit channel at the block length $N_u = 2^{n_u} \leq N = 2^n$: $W_{N_u}^{(i_u)}$ (here, $i_u - 1 = (i_{\mathcal{Z}_u})_b$). For the lower part, there is similarly a new bit channel. For this specific example, $n_u = 5$ and $n_l = 3$. One of the two new bit channels can be determined: $W_{N_l}^{(j_l)} \preceq_1 W_{N_l}^{(i_l)}$. Only the relationship between the two upper channels cannot be determined.

For a given BMS channel W, if the upper bit channels are $W_{N_u}^{(j_u)} \preceq W_{N_u}^{(i_u)}$ by any sorting algorithm, then the following proposition of general partial order (GPO) can be used to determine more intrinsic relationships among bit channels.

Proposition 2 *General partial order (GPO): For a given underlying channel W, consider bit channels i and j $(1 \leq i, j \leq N)$ with $i - 1 = (i_n, i_{n-1}, ..., i_1)_b$ and $j - 1 = (j_n, j_{n-1}, ..., j_1)_b$. Divide the binary expansions of $i - 1$ and $j - 1$ into two parts: $i - 1 = (i_{\mathcal{Z}_u}, i_{\mathcal{Z}_l})_b$ and $j - 1 = (j_{\mathcal{Z}_u}, j_{\mathcal{Z}_l})_b$. If the relationship between $W_{N_u}^{(i_u)}$ and $W_{N_u}^{(j_u)}$ is obtained for the given underlying channel W and the relationship between $W_{N_l}^{(i_l)}$ and $W_{N_l}^{(j_l)}$ is channel independent, then*

$$W_{N_u}^{(i_u)} \succeq W_{N_u}^{(j_u)} \text{ and } W_{N_l}^{(i_l)} \succeq W_{N_l}^{(j_l)} \Rightarrow W_N^{(i)} \succeq W_N^{(j)} \quad (13)$$

Proof From [23] (Lemma 4.7), the following is proven:

$$W_N^{(i)} \succeq W'^{(i)}_N \text{ when } W \succeq W' \quad (14)$$

First, let us prove that when $i_l = j_l$, the following statement is true:

$$W_{N_u}^{(i_u)} \succeq W_{N_u}^{(j_u)} \Rightarrow W_N^{(i)} \succeq W_N^{(j)} \quad (15)$$

For notational convenience, denote $Q = W_{N_u}^{(i_u)}$ and $Z = W_{N_u}^{(j_u)}$. Then, the left-hand side of (15) means that channel Z is stochastically degraded with respect to channel Q. From the channel polarization process in [1], it is known that each bit channel goes through n levels of one-step transformation defined in (1) and (2). The two channels Q and Z are the output channel of the first n_u levels of bit channel i and bit channel j, respectively. These two channels are the input to the remaining levels denoted by i_l and j_l. Then, $W_N^{(i)} = Q_{N_l}^{(i_l)}$ and $W_N^{(j)} = Z_{N_l}^{(j_l)}$. Since Z is stochastically degraded with respect to channel Q, then invoking (14), channel $Z_{N_l}^{(j_l)}$ is stochastically degraded with respect to channel $Q_{N_l}^{(i_l)}$. This proves (15).

Now, we prove that when $i_u = j_u$, the following statement holds:

$$W_{N_l}^{(i_l)} \succeq W_{N_l}^{(j_l)} \Rightarrow W_N^{(i)} \succeq W_N^{(j)} \tag{16}$$

In this case, $Q = W_{N_u}^{(i_u)} = W_{N_u}^{(j_u)}$. The final bit channels i and j are obtained as $W_N^{(i)} = Q_{N_l}^{(i_l)}$ and $W_N^{(j)} = Q_{N_l}^{(j_l)}$. Because $W_{N_l}^{(i_l)} \succeq W_{N_l}^{(j_l)}$ and the relationship between $W_{N_l}^{(i_l)}$ and $W_{N_l}^{(j_l)}$ is channel independent (the condition of this proposition), we can conclude that no matter what the condition of Q is, $W_N^{(i)} \succeq W_N^{(j)}$ holds.

Finally, we prove that
$W_{N_u}^{(i_u)} \succeq W_{N_u}^{(j_u)}$ and $W_{N_l}^{(i_l)} \succeq W_{N_l}^{(j_l)} \Rightarrow W_N^{(i)} \succeq W_N^{(j)}$

To prove this statement, let us define an intermediate channel k: $k-1 = (k_{\mathcal{Z}_u}, k_{\mathcal{Z}_l})_b = (i_{\mathcal{Z}_u}, j_{\mathcal{Z}_l})_b$. From (16), we have $W_N^{(i)} \succeq W_N^{(k)}$. From (15), we have $W_N^{(k)} \succeq W_N^{(j)}$. From the transitivity of the degradation relationship [13], it can be concluded that $W_N^{(i)} \succeq W_N^{(j)}$. □

The procedure to implement GPO is provided in Algorithm 2. From the definition of the matrix O in (10), POIs are stored in the lower triangular part (excluding the diagonal). Therefore, in line 12 of Algorithm 2, POI $O_{ij} = 1$ if condition (13) holds and if $i > j$. When condition (13) holds and $i < j$, O_{ij} cannot set to be one as line 12 does, because only the lower triangular part of O is defined. Instead, we put $O_{ji} = -1$ in line 15 to express the same thing: bit channel j is stochastically degraded with respect to bit channel i.

Algorithm 2 Apply the GPO to determine more intrinsic relationships among bit channels.

```
Require:
    $O$;
Ensure:
    $O$;
1:  // We divide a channel into two parts, eg:
2:  // $\underbrace{xx\cdots x}_{n_u}\underbrace{yy\cdots y}_{n_l}$
3:  // $n_u + n_l \leftarrow n$
4:  $N_l \leftarrow 2^{n_l}, N_u \leftarrow 2^{n_u}$
5:  $i_u, j_u \in [1, N_u]$ // two channels from the upper part
6:  $i_l, j_l \in [1, N_l]$ // two channels from the lower part
7:  $i \leftarrow (i_u - 1) \times N_l + i_l$
8:  $j \leftarrow (j_u - 1) \times N_l + j_l$
9:  // the relationship between bit channels $i_u$ and $j_u$
    can be obtained from POs or from calling any sorting algorithm; the relationship between bit channels $i_l$
    and $j_l$ is obtained from POs
10: if $\left(W_{N_u}^{(i_u)} \succ W_{N_u}^{(j_u)}\right)$ and $\left(W_{N_l}^{(i_l)} \succeq W_{N_l}^{(j_l)}\right)$ then
11:     if $i > j$ then
12:         $O_{ij} \leftarrow 1$
13:     else
14:         // bit channel $j$ is degraded with respect to bit
            channel $i$
15:         $O_{ji} \leftarrow -1$
16:     end if
17: end if
18: return $O$;
```

Remark 3 *The GPO obtained from Proposition 2 works for a given underlying channel W and can be used to order the bit channels constructed from W. Any existing construction procedure (Tal-Vardy's* [13]*, GA* [14–16]*, or density evolution* [12]*) can be called to order the bit channels at level $n_u \leq n$. For example, if the underlying channel W is a binary symmetric channel (BSC), GPO works by first obtaining the relationships determined from POs (Algorithm 1). Then, n is divided as $n = n_u + n_l$. Any sorting algorithm can be applied at level n_u to sort the bit channels in the range of $[1, N_u = 2^{n_u}]$. In this case, Tal-Vardy's merging procedure can be applied. The channel-dependent sorting at the level n_u is combined with the PO ordering at level n_l to produce more relationships among bit channels at level n. In this case, GA does not work since the underlying channel is not an AWGN channel.*

Remark 4 *The procedures of the GPO are not channel dependent: they are universal for all supported channel types. But the ordering obtained from the GPO is dependent on W; therefore, the resulted ordering obtained from the GPO is not universal, such as that of the POs. For any given channel W, GPO can be used to reduce the construction complexity of polar codes because of the operation at a smaller level n_u instead of level n. It can be easily shown that the relationship obtained from Proposition 2 is also transitive. The final K best bit channels that are selected when applying GPO are the same as calling any sorting algorithm* [12–16]*: In a GPO, the upper bit channels are determined from the given sorting algorithm, and the lower part is channel independent.*

3.3 Indirect relationships

For the given underlying channel W, the direct use of GPO and POs sometimes cannot determine the relationship between two bit channels. Instead, some relationships can be obtained from several intermediate channels. Here, is an example. For $N = 2^9$, consider the bit channels $i - 1 = (100110000)_b$ and $j - 1 = (000111001)_b$. Applying the two POs and GPO yields no decision regarding whether $W_N^{(i)}$ is stochastically upgraded or degraded with respect to $W_N^{(j)}$. Let us take a look at a bit channel $k - 1 = (011000001)_b$. We can determine that bit channel i is stochastically upgraded with respect to bit channel k for an AWGN channel with SNR = 1 dB because bit channel “1001” is stochastically upgraded with respect to bit channel “0110” from the Tal-Vardy algorithm and bit channel “10000” is statistically upgraded with respect to bit channel “00001” from the POs. We can determine that bit channel k is stochastically upgraded with respect to bit channel j because bit channel “011000” is stochastically upgraded with respect to bit channel “000111” from the Tal-Vardy algorithm for the same AWGN channel and bit channel “001” is the same as bit channel “001.” Therefore, we have $W_N^{(i)} \succeq W_N^{(k)}$ and $W_N^{(k)} \succeq W_N^{(j)}$, resulting in $W_N^{(i)} \succeq W_N^{(j)}$.

If the relationship between bit channel i and bit channel j can be obtained from one intermediate bit channel, we must be able to obtain the relationship between bit channel i and bit channel j. There can be several intermediate bit channels between bit channel i and j before we can obtain the relationship between bit channel i and bit channel j. Let us refer to these POIs as indirect relationships. Proposition 3 can identify all indirect relationships in the matrix O after applying the POs and GPO.

Proposition 3 *For a bit channel k $(1 \leq k \leq N)$, if there are bit channels i and j (whose relationship is undetermined from POs and GPO) such that $W_N^{(i)} \succeq W_N^{(k)}$ and $W_N^{(k)} \succeq W_N^{(j)}$, then $W_N^{(i)} \succeq W_N^{(j)}$ is newly identified. When k is searched from 1 to N and i and j is also looped from 1 to N under the loop of k, then ALL indirect relationships between ALL bit channels can be identified.*

Proof The first half of this proposition is the direct use of the transitive property of the POs and GPO. According to the second half of the proposition, when the intermediate channel k goes from 1 to N and the bit channels i and j are also looped from 1 to N (within the loop of k), all indirect relationships among all bit channels can be identified. In the following, the second part of this proposition is proven.

For the existing POIs determined from the POs and GPO (stored in O), pick any two bit channels i and j $(1 \leq i, j \leq N)$ whose relationship is undetermined: $O_{ij} = 0$ and $O_{ji} = 0$. However, from the relationship matrix O, the following relationships can be determined:

$$W_N^{(i)} \succeq W_N^{(k_1)} \succeq W_N^{(k_2)} \succeq \cdots \succeq W_N^{(k_m)} \succeq W_N^{(j)} \tag{17}$$

where $1 \leq k_1, k_2, ..., k_m \leq N$. Let $\mathcal{K} = \{k_1, k_2, ..., k_m\}$. Define $k_l = \min(\mathcal{K})$. When k goes from 1 to N, the first transitive relationship $W_N^{(k_{l-1})} \succeq W_N^{(k_l)} \succeq W_N^{(k_{l+1})}$ is encountered. Then, the POI between bit channel k_{l-1} and k_{l+1} can be updated in the matrix O. The remaining set of $\mathcal{K}$ becomes $\mathcal{K} = \{1, 2, ..., k_{l-1}, k_{l+1}, ..., k_m\}$ with the corresponding relationship

$$W_N^{(i)} \succeq \cdots \succeq W_N^{(k_{l-1})} \succeq W_N^{(k_{l+1})} \succeq \cdots \succeq W_N^{(j)} \tag{18}$$

Compared with the relationship in (17), the relationship in (18) has one less bit channel $W_N^{(k_l)}$. The transitive relationship $W_N^{(k_{l-1})} \succeq W_N^{(k_l)} \succeq W_N^{(k_{l+1})}$ is replaced by the direct relationship $W_N^{(k_{l-1})} \succeq W_N^{(k_{l+1})}$. This process continues with one new direct POI updated in each step until the last step, where $W_N^{(i)} \succeq W_N^{(j)}$ is reached. As the intermediate channel k goes from 1 to N and i and j also goes from 1 to N within the loop of k, any indirect relationship between bit channel i and j $(1 \leq i, j \leq N)$ can be identified during the three loops. □

3.4 GPO parameter optimization

For the GPO, a parameter to be determined is the upper level n_u. In this paper, the sorting algorithm in [13] is employed. The parameter n_u is estimated based on this sorting algorithm. In the lth $(1 \leq l \leq n)$ approximation stage, different calculations are called according to “0” or “1” of the lth bit of the binary expansion of a bit channel. Denote t_0 and t_1 as the calculation time that corresponds to 0 and 1. Then, the calculating time for any channel is $x \times t_0 + y \times t_1$ (x, y is the number of 0s and 1s of the binary expansion of this channel). Assume that the average Hamming weight of the channels in $\mathcal{U}$ is w_u. From the two POs, we can easily show that the three bit channels with the smallest indices are the worst, and the three bit channels with the largest indices are the best. Therefore, the ordering of these six bit channels can be saved for all sorting algorithms. The approximate calculation time to sort the bit channels at level n_u and to sort the bit channels at $\mathcal{U}$ is:

$$T_{n_u} = \frac{(N_u - 6)n_u}{2}(t_0 + t_1) + K_u(n - w_u)t_0 + K_u w_u t_1 \tag{19}$$

where $K_u = |\mathcal{U}|$ is cardinality of the set $\mathcal{U}$.

From (19), the calculation time of GPO is dependent on the size of the set $\mathcal{U}$, K_u, and the average Hamming weight of the binary expansion of the elements in $\mathcal{U}$. Therefore, a theoretical estimation of the optimal selection of n_u is unavailable. However, any selection of $3 \leq n_u \leq N - 1$ can achieve an advantage in terms of the construction time given the nature of the GPO. In Section 4, a quasi-analytical estimation of n_u for n up to 10, which produces the most efficient construction of polar codes, is performed. Estimation of n_u with larger block lengths is readily achievable.

3.5 Outline of the overall GPO algorithm

Algorithm 3 is a top-level algorithm that implements PO $\preceq_1$, PO $\preceq_2$, GPO, and the indirect relationships presented in Proposition 3.

The original POI matrix O is obtained from Algorithm 1 (PO $\preceq_1$, PO $\preceq_2$). Then, the POI matrix O is updated by Algorithm 2 (GPO). As stated in Proposition 3, some intrinsic relationships exist between bit channels that cannot be obtained from POs and GPO, but can be obtained from several intermediate channels. This type of intrinsic relationship is obtained from Proposition 3, whose algorithm is clear; therefore, it is omitted here.

Once the POI matrix O is updated by Proposition 3 (line 7 of Algorithm 3), we can obtain the two vectors v_1^N and w_1^N: the entry v_i of v_1^N corresponds to the number of bit channels that are stochastically degraded with respect to bit channel i, and the entry w_j of w_1^N corresponds to the number of bit channels that are stochastically upgraded with respect to bit channel j (line 11 of Algorithm 3). Then, Proposition 1 can be applied to assign elements to the sets $\mathcal{I}$, $\mathcal{F}$, and $\mathcal{U}$ (line 12-13 of Algorithm 3).

4 Numerical results

In this section, the efficiency of applying POs and GPO is quantified. The underlying channel is the AWGN with a SNR of 1 dB. When applying the GPO, the algorithm called to sort the lower dimension channels is from [13]. We use PO to refer to the results obtained after applying the two POs. The legend PO+GPO indicates that POs (Algorithm 1) and GPO (Algorithm 2) are all employed. If indirect relationships (IR) from Proposition 3 is also applied after PO and GPO, the legend is PO+GPO+IR. All construction results in this section match the constructions in [13]. Therefore, in the following, we show only the efficiency results.

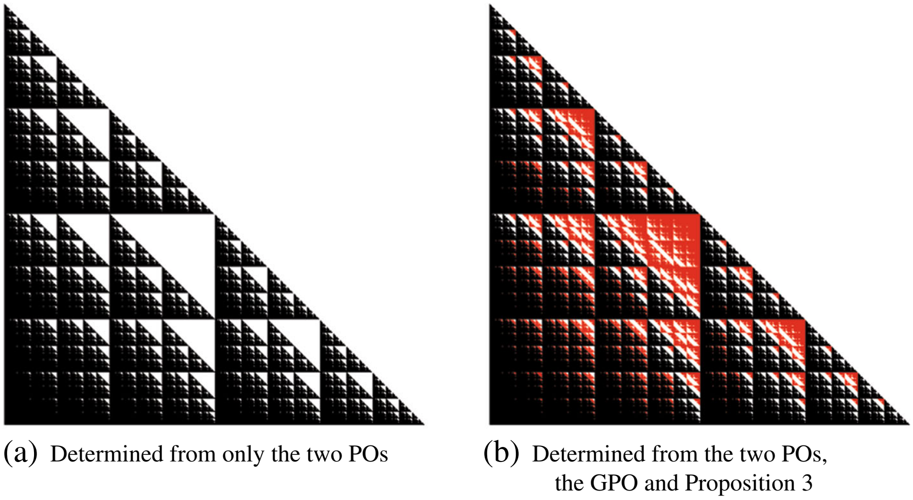
Algorithm 3 Determine elements of $\mathcal{I}, \mathcal{F}, \mathcal{U}$.

```
Require:
    $n, R, snr$;
Ensure:
    $\mathcal{I}, \mathcal{F}, \mathcal{U}$;
1:  $N \leftarrow 2^n, K \leftarrow \lfloor N \times R \rfloor, N_f \leftarrow N - K$
2:  $n_u \leftarrow n - 3, N_u \leftarrow 2^{n_u}$
3:  // Implementation of the two POs: PO $\preceq_1$, PO $\preceq_2$
4:  Call Algorithm 1, return the matrix $O$ on the basis of $n$.
5:  // Implementation of GPO
6:  Call Algorithm 2, return the matrix $O$.
7:  Process $O$ according to Proposition 3 to obtain additional POIs.
8:  vector $s_1^N$: $s_i$ the number of bit channels degraded with respect to bit channel $i$
9:  vector $f_1^N$: $f_i$ the number of bit channels upgraded with respect to bit channel $i$
10: // this function counts the number of channels degraded with respect to each bit channel and upgraded with respect to each bit channel
11: $(s_1^N, f_1^N) \leftarrow$ counting channels from $O$
12: $\mathcal{I} \leftarrow find\,(s_1^N \geq N_f)$
13: $\mathcal{F} \leftarrow find\,(f_1^N \geq K)$
14: Put elements not in $\mathcal{I}$ and $\mathcal{F}$ to $\mathcal{U}$
```

To obtain the first impression of how well PO $\preceq_1$, PO $\preceq_2$, GPO, and indirect relationships work, the elements of the matrix O are plotted in Fig. 1, where a black or a red dot at O_{ij} denotes that the relationship between bit channel i and bit channel j is determined, and the white space indicates that the corresponding relations are not determined. In Fig. 1, the polar code has the block length $N = 2^9$, and the underlying channel is AWGN with a SNR of 1 dB. In Fig. 1a, the black dots indicate that the relationship is determined from the two POs, and the red dots in Fig. 1b indicate the additional intrinsic relationships obtained from the GPO and the indirect relationships. The application of GPO and indirect relationships fills some portion of the white space in Fig. 1a, which indicates that additional relations are identified.

Before quantifying the savings of GPO and IR (here, we use IR to refer to the results obtained from Proposition 3), the parameter n_u of the GPO needs to be quantified. Let us consider $n = 9$, $N = 2^n = 512$, $R = 0.5$, and SNR $= 1$ dB as an example to show the optimal selection of the dimension n_u. Among all code rates, $R = 0.5$ is the worst case, which produces the largest size of $\mathcal{U}$. When $R = 0.5$, we assume that the set $\mathcal{U}$ has the same property of the entire set $\{1, 2, .., N\}$: the average weight of the binary expansion of the bit channels in $\mathcal{U}$ is $w_u \approx \frac{n}{2}$. From (19), we have $T_{n_u} = \left(\frac{(N_u-6)n_u}{2} + \frac{K_u n}{2}\right)(t_0 + t_1)$. If the upper dimension is $n_u = 6$, $K_u = 104$ elements exist in the undetermined set $\mathcal{U}$ (obtained from the algorithm of POs). Note that the quality of 58 bit channels needs to be calculated on the smaller dimension $n_u = 6$ (removing the largest six bit channels and the smallest six bit channels). In this example, $T_6 = \left(\frac{(64-6)\times 6}{2} + \frac{104\times 9}{2}\right)(t_0+t_1) = 642(t_0 + t_1)$ must be satisfied to select the best K bit channels. As a comparison, the original complete sorting based on $n = 9$ requires $T_9 = 2^9 \times \frac{9}{2}(t_0 + t_1) = 2304(t_0 + t_1)$ to select the K best bit channels. Table 1 lists the theoretical calculation times based on different dimensions n_u. The expression $n_u = 6$ produces the most efficient construction of polar codes. Table II lists the comparison of the theoretical calculation time and the real implementation time relative to the complete construction of polar codes. In Table 2, the ratio is T_{n_u}/T_n. As shown in Table 2, the approximation of the calculation time in (19) is similar for $R = 0.5$.

Note that the running time of POs in Algorithm 1 can be negligible compared with the running time of the construction of [13] because Algorithm 1 involves only simple operations (no real multiplications). With the relationship matrix O obtained from Algorithm 1, the set $\mathcal{U}$ can be determined. Then, a theoretical estimation of an optimal


(a) Determined from only the two POs

(b) Determined from the two POs, the GPO and Proposition 3

Fig. 1 The relationship of all bit channels with $N = 2^9$. **a** Black dots indicate that the relationship is determined from the two POs whereas white dots indicate that the relationship cannot be determined from the two POs. **b** The underlying channel is AWGN with a SNR of 1 dB. On top of the black dots, the red dots indicate the additional intrinsic relationships obtained from the GPO

n_u can be obtained based on T_{n_u} in (19). In this section, all examples consider $n_u = n-3$, which works for n up to 10.

The size $K_u = |\mathcal{U}|$ determines the remaining calculations of the sorting algorithms. For example, K_u determines the number of callings of the Tal-Vardy algorithm [13]. In Fig. 2, the ratio $\gamma = K_u/N$ is plotted for $N = 2^9$ as a function of the code rate R (the solid circled line), which shows that the combinations of the two POs can determine 91% of the bit channels with $\gamma = 9\%$ for $R = 0.1$. For $R = 0.5$, which is the worst case, the two POs determine 50% of the bit channels, which is still a significant saving, considering the complexity of the sorting algorithms [12, 13]. A small γ is desired since it indicates a low sorting effort. The dash-dotted line in Fig. 2 is γ when applying POs and GPO (without running the search for the indirect relationships). The legend for this line is "PO+GPO." The dotted line in Fig. 2 is γ after applying POs, GPO, and Proposition 3 (searching for indirect relationships). The gap between PO+GPO and PO+GPO+IR shows that after running PO+GPO, there are indirect relationships imbedded in the POI matrix O. As shown in Fig. 2, the value of γ decreases with R when $R > 0.5$ and increases with R when $R < 0.5$. At the code rate R = 0.5, all three lines (PO, PO+GPO, and PO+GPO+IR) have the largest value of γ. However, the savings of applying GPO and Proposition 3 is the largest at $R = 0.5$. For the small code rate $R \leq 0.1$, less than 10% of the remaining channels are in the undetermined set $\mathcal{U}$.

Figure 3 shows the relationship between γ and the block length for two fixed code rates $R = 0.5$ and $R = 0.1$. The three circled lines represent $R = 0.5$: the solid line denotes POs, the dashed-dotted line denotes PO+GPO, and the dotted line is PO+GPO+IR. Similarly, the three lines with stars are values of γ for $R = 0.1$. As shown in Fig. 3, the number of undetermined channels slowly increases with the block length when applying POs. For PO+GPO and PO+GPO+IR, the trend is the opposite: the portion of undetermined bit channels decreases with the block length. This is caused by the following facts: with the increase of the block length $N = 2^n$, there are more positions for bits 0 and 1 to choose from, producing more combinations of bit channels whose binary expansions cannot be characterized by the two POs. This in turn causes the increase of the size of the undetermined set $\mathcal{U}$, leading to a slow increase of γ when applying POs only, as seen in Fig. 2. By applying GPO, these undetermined relationships are revealed from calling any existing sorting procedure (Tal-Vardy's algorithm in our work).

Table 1 The theoretical time of the $n = 9$ for different n_u

n_u	4	5	6	7	9
K_u	212	150	104	68	0
$T_{n_u}/(t_0 + t_1)$	974	740	642	733	2304

Table 2 The ratio of T_{n_u}/T_n to select the K best bit channels for $n = 8, n = 9$ and $n = 10$

n	Simulated ratio	Theoretical ratio
8	0.307	0.306
9	0.286	0.279
10	0.270	0.262

For all values of n, n_u is set to be $n_u = n - 3$ and $R = 0.5$. The theoretical ratio is based on the approximation in (19)

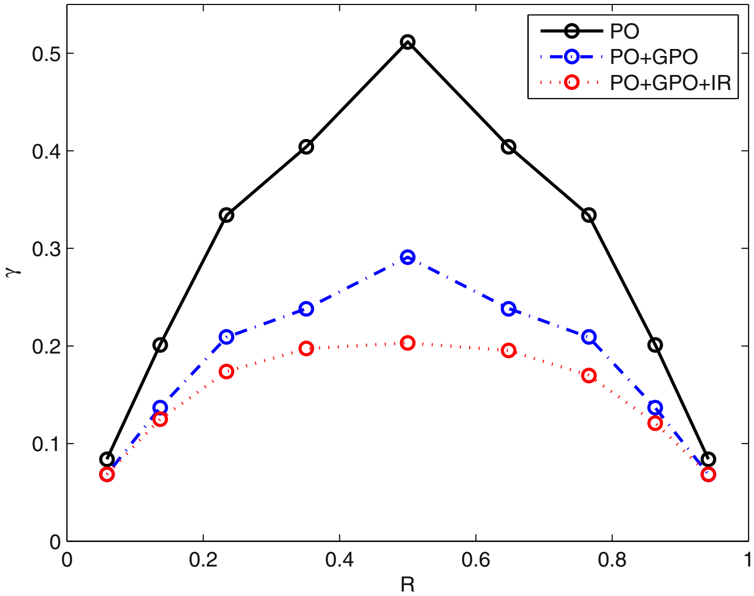


Fig. 2 Value of γ as a function of the code rate R with POs and GPO+PO. The underlying channel is AWGN with a SNR of 1 dB. The block length is $N = 2^9$

Therefore, the portion of relationships being newly discovered increases with n, resulting in a decreasing γ with n. From Fig. 3, it can be seen that GPO+PO+IR can produce approximately 82% of the savings in terms of the computation time for polar code construction with a code rate $R = 0.5$ and $N = 1024$.

5 Conclusion

In this paper, partial orders of polar codes for reducing the construction complexity are investigated. Detailed quantification of the existing partial orders, which displays a clear potential of combining partial orders in the construction of polar codes, is provided. To further reduce the computation complexity, a general partial order, which works at a lower block length than the given block length, is proposed. In addition to the relationships obtained from the existing partial orders and the proposed general partial order, indirect relationships are also discussed in the paper. Detailed algorithms for implementing all the partial orders and finding indirect relations are provided to

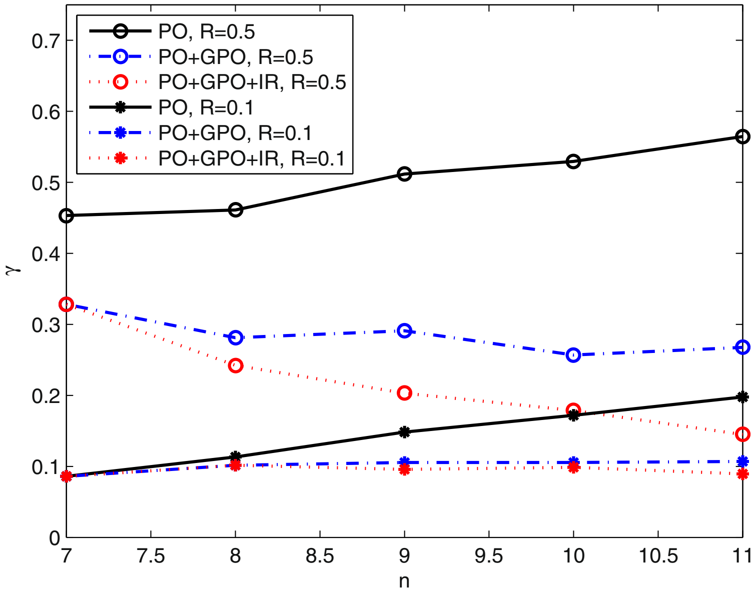


Fig. 3 Value of γ with two different code rates: $R = 0.1$ and $R = 0.5$. The underlying channel is AWGN with a SNR of 1 dB

quantify their savings. Substantial savings are observed when combining all the partial orders in the paper for polar code construction.

Abbreviations

AWGN: Additive white Gaussian noise; BEC: Binary erasure channels; BMS: Binary-input, memoryless, output-symmetric; GA: Gaussian approximations; GPO: General partial order; IR: Indirect relationships; PO: Partial order; POI: Partial order indicator; SC: Successive cancelation

Funding

This work was supported in part by National Natural Science Foundation of China through grant 61501002, in part by Natural Science Project of Ministry of Education of Anhui through grant KJ2015A102, and in part by Talents Recruitment Program of Anhui University.

Availability of data and materials

Not applicable.

Authors' contributions

LL proposed the polar code construction reduction in the paper. WW implemented it and proved the theorems in the paper. KN provided format and logic structures of the paper. All authors read and approved the final manuscript.

Authors' information

Liping Li is now an associate professor of the Key Laboratory of Intelligent Computing and Signal Processing of the Ministry of Education of China, Anhui University, Hefei, China. She got her PhD in Dept. of Electrical and Computer Engineering at North Carolina State University, Raleigh, NC, USA, in 2009. Her current research interest is in channel coding, especially polar codes. Dr. Li's research topic during her PhD studies was multiple-access interference analysis and synchronization for ultra-wideband communications. Then she worked on a LTE indoor channel sounding and modeling project in University of Colorado at Boulder, collaborating with Verizon. From 2010 to 2013, she worked at Maxlinear Inc. as a staff engineer in the communication group. At Maxlinear, she worked on SoC designs for the ISDB-T standard and the DVB-S standard, covering modules on OFDM and LDPC. In Sept. 2013, she joined Anhui University and started her research on polar codes until now.

Wei Wang is currently pursuing the B.S. degree in the Key Laboratory of Intelligent Computing and Signal Processing of the Ministry of Education of China, Anhui University and School of Electronics and Information Engineering, Anhui University. His research interest is in polar codes. Specifically, his research is to reduce the complexity of polar code construction. He was awarded the first prize of the Seventh National College Students Mathematics Competition in 2015. He was awarded the third prize of the thirtieth national middle school physics competition in Anhui Province in 2013. He was awarded the national inspirational scholarship for 2014 and 2015 academic year. He also achieved the second prize of the Ninth International Students' Innovation and Entrepreneurship Competition in Anhui Province (I CAN) in 2016.

Kai Niu received the B.S. degree in information engineering and Ph.D. degree in signal and information processing from Beijing University of Posts and Telecommunications (BUPT), Beijing, China, in 1998 in 2003, respectively. Currently, he is a professor at the School of Information and Communication Engineering of BUPT. His research interests are in the area of channel coding and broadband wireless communication, particularly on the practical design of polar codes and study of polar decoding algorithms.

Competing interests

The authors declare that they have no competing interests.

Publisher's Note

Author details

[1]Key Laboratory of Intelligent Computing and Signal Processing of the Ministry of Education of China, Anhui University, Hefei, People's Republic of China. [2]Key Laboratory of Universal Wireless Communication, Ministry of Education, Beijing University of Posts and Telecommunications, Beijing 100876, People's Republic of China.

Received: 17 December 2017 Accepted: 11 December 2018
Published online: 14 January 2019

References

1. E. Arıkan, Channel polarization: a method for constructing capacity-achieving codes for symmetric binary-input memoryless channels. IEEE Trans. Inf. Theory. **55**(7), 3051–3073 (2009)
2. E. Arıkan, I. E. Telatar, in *IEEE International Symposium on Information Theory (ISIT)*. On the rate of channel polarization, (2009), pp. 1493–1495
3. S. B. Korada, E. Sasoglu, R. Urbanke, Polar codes: characterization of exponent, bounds, and constructions. IEEE Trans. Inf. Theory. **56**(12), 6253–6264 (2010)
4. S. H. Hassani, R. Mori, T. Tanaka, R. L. Urbanke, Rate-dependent analysis of the asymptotic behavior of channel polarization. IEEE Trans. Inf. Theory. **59**(4), 2267–2276 (2013)
5. N. Hussami, S. Korada, R. Urbanke, in *IEEE International Symposium on Information Theory (ISIT)*. Performance of polar codes for channel and source coding, (2009), pp. 1488–1492
6. A. Eslami, H. Pishro-Nik, in *48th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*. On bit error rate performance of polar codes in finite regime, (2010), pp. 188–194
7. J. Guo, M. Qin, A. G. i Fabregas, P. H. Siegel, in *2014 IEEE International Symposium on Information Theory Proceedings (ISIT)*. Enhanced belief propagation decoding of polar codes through concatenation, (2014), pp. 2987–2991
8. U. U. Fayyaz, J. R. Barry, in *2013 IEEE International Conference on Communications (ICC)*. Polar codes for partial response channels, (2013), pp. 4337–4341
9. E. Sasoglu, E. Telatar, E. Arikan, in *IEEE Information Theory Workshop(ITW)*. Polarization for arbitrary discrete memoryless channels, (2009)
10. R. Mori, T. Tanaka, in *IEEE Information Theory Workshop(ITW)*. Non-binary polar codes using Reed-Solomon codes and algebraic geometry codes, (2010), pp. 1–5
11. A. G. Sahebi, S. S. Pradhan, in *2011 49th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*. Multilevel polarization of polar codes over arbitrary discrete memoryless channels, (2011), pp. 1718–1725
12. R. Mori, T. Tanaka, Performance of polar codes with the construction using density evolution. IEEE Commun. Lett. **13**(7), 519–521 (2009)
13. I. Tal, A. Vardy, How to construct polar codes. IEEE Trans. Inf. Theory. **59**(10), 6562–6582 (2013)
14. P. Trifonov, Efficient design and decoding of polar codes. IEEE Trans. Commun. **60**(11), 3221–3227 (2012)
15. D. Wu, Y. Li, Y. Sun, Construction and block error rate analysis of polar codes over AWGN channel based on Gaussian approximation. IEEE Commun. Lett. **18**(7), 1099–1102 (2014)
16. J. Dai, K. Niu, Z. Si, J. Lin, *Evaluation and optimization of Gaussian approximation for polar codes*, (2016). https://arxiv.org/abs/1511.07236
17. C. Schurch, in *IEEE International Symposium on Information Theory (ISIT)*. A partial order for the synthesized channels of a polar code, (2016), pp. 220–224
18. M. Bardet, V. Dragoi, A. Otmani, J.-P. Tillich, *Algebraic properties of polar codes from a new polynomial formalism*, (2016). https://arxiv.org/abs/1601.06215
19. M. Mondelli, S. H. Hassani, R. Urbanke, in *IEEE International Symposium on Information Theory (ISIT)*. Construction of polar codes with sublinear complexity, (2017)
20. G. He, J.-C. Belfiore, I. Land, G. Yang, X. Liu, Y. Chen, R. Li, J. Wang, Y. Ge, R. Zhang, W. Tong, *β-expansion: a theoretical framework for fast and recursive construction of polar codes*, (2017). https://arxiv.org/abs/1704.05709
21. W. Wang, L. Li, in *IEEE International Wireless Communications and Mobile Computing Conference*. Efficient construction of polar codes, (2017), pp. 1594–1598
22. T. Richardson, R. Urbanke, *Modern coding theory*. (Cambridge University Press, Cambridge, 2008)
23. S. B. Korada, *Polar codes for channel and source coding. PhD Thesis*, (2009)