Technical University of Crete

School of Electrical and Computer Engineering

Information bit selection in Polar Coding for the Binary Symmetric Channel

Author

Konstantinos Bountrogiannis

A thesis submitted in partial fulfillment of
the requirements for the degree of Diploma in
Electrical and Computer Engineering

Supervisor

Assoc. Prof. George N. Karystinos

Commitee Members

Assoc. Prof. Aggelos Bletsas

Prof. Athanasios P. Liavas

Chania 2018

Abstract

Polar codes is a new scheme of channel coding, which is the first provably capacity-achieving coding scheme for a wide class of channels, the binary discrete memoryless channels. At the same time, they use low complexity encoding and decoding algorithms, which makes them attractive for a wide range of use-cases. These algorithms scale as $O(NlogN)$, where N is the blocklength of the code. Polar codes exploit channel polarization, a very common phenomenon which arises when one takes N independent copies of a channel and transforms them into another set of N channels. Under channel polarization, the channels are converted to a set of extremal (either perfect or completely noisy) channels, called bit-channels. In the presence of channel polarization, the information vector is sent through the perfect bit-channels, while a fixed vector of arbitrary bits is sent through the useless bit-channels. A problem which arises is the determination of which bit-channels are perfect and which are useless. This problem, called the "construction of polar codes" among researchers, has been addressed successfully and efficiently only for the binary erasure channel (BEC). The non-universality property of polar codes complicates their construction, because the behaviour of a bit-channel may be perfect for one physical channel but noisy for another. The adoption of polar codes in 5G NR strengthens the demand for a fast and adaptive construction scheme. This thesis attempts to design an efficient algorithm for the construction of polar codes for the binary symmetric channel (BSC), taking advantage of proved universal partial orders among the bit-channels and state-of-the-art algorithms which approximate efficiently upper and lower bounds of the probability of error of the bit-channels. The simulation results show a marginal time-running difference over the explicit use of the approximation algorithms, which can be used computationally for a more accurate construction.

Περίληψη

Οι πολικοί κώδικες *(polar codes)* είναι μια σύγχρονη μέθοδος κωδικοποίησης καναλιού, η πρώτη μέθοδος που αποδεδειγμένα επιτυγχάνει την χωρητικότητα του καναλιού για μια μεγάλη κατηγορία καναλιών, τα δυαδικά διακριτά κανάλια χωρίς μνήμη. Την ίδια στιγμή, χρησιμοποιούν αλγορίθμους κωδικοποίησης και αποκωδικοποίησης χαμηλής πολυπλοκότητας, κάτι που τους κάνει ελκυστικούς για πολλές χρήσεις. Οι αλγόριθμοι αυτοί έχουν πολυπλοκότητα της τάξης $O(NlogN)$, όπου N είναι το μήκος μπλοκ του κώδικα. Οι πολικοί κώδικες αξιοποιούν ένα φαινόμενο που ονομάζεται πόλωση καναλιού *(channel polarization)*, ένα σύνηθες φαινόμενο που προκύπτει όταν μετασχηματίζουμε N ανεξάρτητα αντίγραφα ενός καναλιού σε ένα άλλο σύνολο από N κανάλια. Τα κανάλια πολώνονται, με την έννοια ότι μετατρέπονται σε ένα σύνολο από ακραία κανάλια (είτε τέλεια είτε εντελώς θορυβώδη), τα οποία ονομάζουμε bit-channels. Υπό την παρουσία της πόλωσης καναλιού, η πληροφορία αποστέλλεται μέσα από τα τέλεια bit-channels, ενώ μέσα από τα άχρηστα bit-channels αποστέλλεται μια αυθαίρετη στατική ακολουθία από bits. Ένα πρόβλημα που προκύπτει είναι η εξακρίβωση των bit-channels που είναι τέλεια και αυτών που είναι άχρηστα. Αυτό το πρόβλημα, που από τους ερευνητές ονομάζεται "κατασκευή των πολικών κωδίκων", έχει επιλυθεί με γρήγορο τρόπο μόνο για το δυαδικό κανάλι διαγραφής (BEC). Το γεγονός ότι τα bit-channels των πολικών κωδίκων δεν έχουν ενιαία συμπεριφορά για όλα τα φυσικά κανάλια στα οποία κατασκευάζεται ο πολικός κώδικας, περιπλέκει το πρόβλημα διότι ένα bit-channel μπορεί να είναι τέλειο για ένα πολικό κώδικα αλλά θορυβώδες για έναν άλλον. Η αξιοποίηση των πολικών κωδίκων στο 5G NR ενισχύει την ανάγκη για έναν γρήγορο και ευπροσάρμοστο αλγόριθμο κατασκευής. Αυτή η διπλωματική εργασία προσπαθεί να σχεδιάσει έναν αποδοτικό αλγόριθμο για την κατασκευή των πολικών κωδίκων για το δυαδικό συμμετρικό κανάλι (BSC), αξιοποιώντας κάποιες μερικές διατάξεις (partial orders) μεταξύ των bit-channels που έχουν αποδειχθεί ότι ισχύουν για όλους τους πολικούς κώδικες, και κάποιους σύγχρονους αλγορίθμους που εκτιμούν αποδοτικά άνω και κάτω φράγματα της πιθανότητας σφάλματος των bit-channels. Τα αποτελέσματα των προσομοιώσεων δείχνουν μια σημαντική διαφορά στην ταχύτητα του προτεινόμενου αλγορίθμου από την αποκλειστική χρήση των προσεγγιστικών αλγορίθμων, τέτοια ώστε ο χρόνος που εξοικονομείται να μπορεί να αξιοποιηθεί υπολογιστικά για μια πιο ακριβή κατασκευή του κώδικα.

Θα ήθελα να ευχαριστήσω τον καθηγητή μου, κ. Γ. Καρυστινό για την συνεχή στήριξη που μου έδωσε καθ΄ όλη την διάρκεια της εκπόνησης αυτής της εργασίας. Θα ήθελα επίσης να ευχαριστήσω όλους τους καθηγητές μου, ιδιαίτερα τους κ.κ. Α. Λιάβα και Α. Μπλέτσα, για τις γνώσεις και τα δημιουργικά αιρεθίσματα που μου έδωσαν κατά την διάρκεια των σπουδών μου.

Επίσης ευχαριστώ όλους τους φίλους μου, για όλες τις ωραίες στιγμές που περάσαμε μαζί.

Τέλος, αφιερώνω αυτήν την εργασία στην οικογένειά μου, τους γονείς και τα αδέρφια μου, που με στήριξαν καθ΄ όλη την διάρκεια των σπουδών μου.

Contents

1 Channel Polarization

Channel polarization, proposed by Arikan in [1], is a method where one takes N independent copies of a given binary discrete memoryless channel (B-DMC) W and transforms them into another set of N binary-input channels $\{W_N^{(i)} : 1 \leq i \leq N\}$ with the property that, as N becomes large, the symmetric capacities $\{I_N^{(i)}\}$ become either 0 or 1 with probability 1 and also $\sum_i I_N^{(i)} = NI(W)$.

1.1 Preliminaries

Given a B-DMC $W : X \to Y$, two parameters are of great significance in the following analysis, the symmetric capacity

$$I(W) \triangleq \sum_{y \in Y} \sum_{x \in X} \frac{1}{2} W(y|x) \log \frac{W(y|x)}{\frac{1}{2}W(y|0) + \frac{1}{2}W(y|1)} \tag{1}$$

and the Bhattacharyya parameter

$$Z(W) \triangleq \sum_{y \in Y} \sqrt{W(y|0)W(y|1)}. \tag{2}$$

The parameter $I(W)$ is a measure of *rate* and it is easy to notice that it is the channel capacity when W is symmetric. On the other hand, $Z(W)$ is a measure of *reliability*, since it is an upper bound on the probability of error of maximum-likelihood decoding when W is used only once to transmit a bit. Indeed, consider a B-DMC W and the error event under ML decoding $\varepsilon \doteq \{(x, y) \in X \times Y : \frac{W(y|x \oplus 1)}{W(y|x)} \geq 1\}$. Then, the probability of error is

$$\begin{aligned}
P(\varepsilon) &= \sum_{y \in Y} \sum_{x \in X} P_{X,Y}(x, y) \mathbb{1}_{\mathcal{E}}(x, y) = \sum_{y \in Y} \sum_{x \in X} P_X(x) P_{Y|X=x}(y|x) \mathbb{1}_{\mathcal{E}}(x, y) \\
&= \sum_{y \in Y} \sum_{x \in X} P_X(x) W(y|x) \mathbb{1}_{\mathcal{E}}(x, y) \leq \sum_{y \in Y} \sum_{x \in X} P_X(x) W(y|x) \sqrt{\frac{W(y|x \oplus 1)}{W(y|x)}} \\
&= \sum_{y \in Y} \sum_{x \in X} P_X(x) \sqrt{W(y|x)W(y|x \oplus 1)} = \sum_{y \in Y} \sum_{x \in X} P_X(x) \sqrt{W(y|0)W(y|1)} \\
&= \sum_{y \in Y} \sqrt{W(y|0)W(y|1)} = Z(W).
\end{aligned} \tag{3}$$

Both $I(W)$ and $Z(W)$ take values in $[0, 1]$. The two parameters are related with the following two bounds.

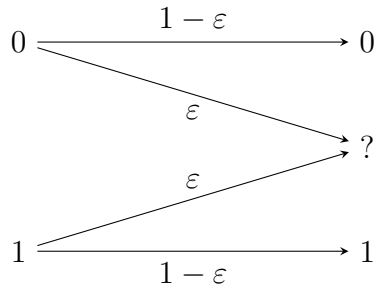

Figure 1: The binary erasure channel (BEC) with erasure probability ε.

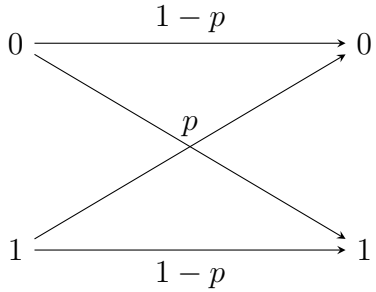

Figure 2: The binary symmetric channel (BSC) with crossover probability p.

Proposition 1.

$$I(W) \geq \log \frac{2}{1 + Z(W)}, \tag{4}$$

$$I(W) \leq \sqrt{1 - Z(W)^2}. \tag{5}$$

□

The proof of Proposition 1 is provided in the Appendix.

From (4) and (5), we infer the following corollary.

Corollary 1. *$Z(W) \longrightarrow 1$ iff $I(W) \longrightarrow 0$. Similarly, $Z(W) \longrightarrow 0$ iff $I(W) \longrightarrow 1$.* □

Lastly, we define the two binary-input channels that will be mentioned throughout this presentation of polar codes, the binary erasure channel (BEC) and the binary symmetric channel (BSC). In BEC (Fig. 1), the receiver either receives the bit transmitted correctly, with probability $1 - \epsilon$, or receives a message (an *erasure symbol*) that the bit was *erased* during the transmission, with *erasure probability* ϵ. In BSC (Fig. 2), the receiver either receives the bit transmitted correctly, with probability $1 - p$, or receives the bit *flipped*, with *crossover probability* p. For BEC, (1) and (2) give the following equations

$$\begin{aligned} I(W) &= 1 - \epsilon, \\ Z(W) &= \epsilon, \end{aligned} \tag{6}$$

while, for BSC, they become

$$\begin{aligned} I(W) &= 1 + p \log p + (1 - p) \log(1 - p), \\ Z(W) &= 2\sqrt{p(1 - p)}. \end{aligned} \tag{7}$$

Henceforth, we will use the notation α_1^N to denote the row vector $\alpha = (\alpha_1, \alpha_2, \ldots, \alpha_N)$. Given such a vector α_1^N, we write α_i^j as a shorthand to denote the subvector $(\alpha_i, \ldots, \alpha_j)$

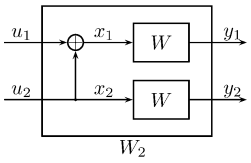

Figure 3: Construction of W_2 from two independent copies of $W_1 = W$.

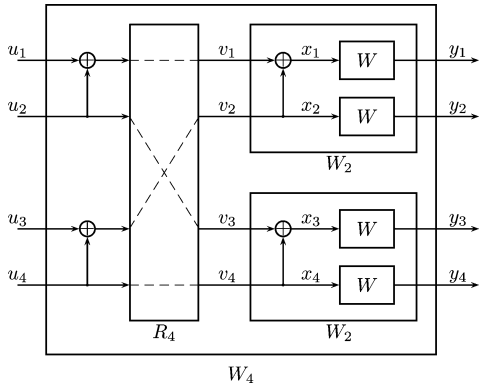

Figure 4: Construction of W_4 from two independent copies of W_2.

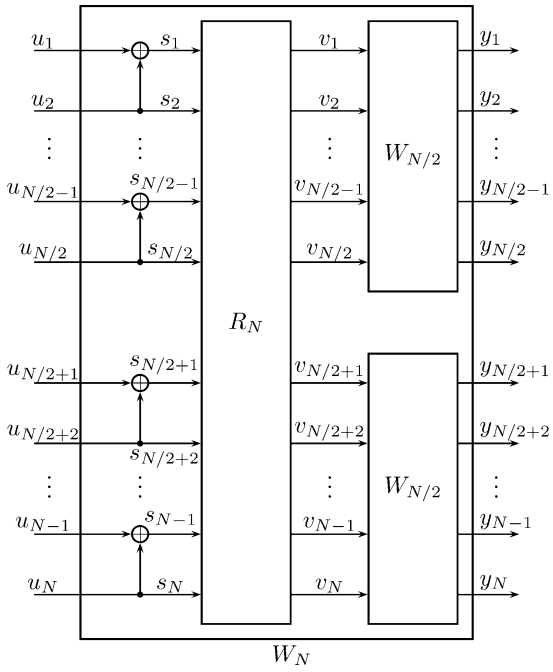

Figure 5: Construction of W_N from two independent copies of $W_{N/2}$.

when $1 \leq i \leq j \leq N$, whereas α_i^j is regarded as void when $j < i$. Given the vector α_1^N and $A \subset \{1, \ldots, N\}$, we write α_A to denote the subvector $(\alpha_k : k \in A)$. We write $\alpha_{i,e}^j$ to denote the subvector of α_i^j with even indices $(\alpha_k : i \leq k \leq j; k \text{ even})$, and $\alpha_{i,o}^j$ to denote the subvector of α_i^j with odd indices $(\alpha_k : i \leq k \leq j; k \text{ odd})$.

1.2 Channel Polarization

In this section we will address the phenomenon of channel polarization, as described above. Shortly afterwards, Polar Codes will be defined and proved to be capacity-achieving, by exploiting this feature.

1.2.1 Channel Combining

The *Channel Combining* phase of channel polarization concerns the recursive transformation of a set of N independent copies of a given B-DMC, namely W^N, into another vector channel $W_N : X^N \to Y^N$, $N = 2^n$, $n \geq 0$. The general recursion step is illustrated in Fig. 5. The initial step of the recursion is $W_1 \triangleq W$. Then, two independent copies of $W_{N/2}$ are combined to produce W_N. The input vector u_1^N of W_N is first transformed into s_1^N such that $s_{2i-1} = u_{2i-1} \oplus u_{2i}$ and $s_{2i} = u_{2i}$ for

$1 \leq i \leq N/2$. The vector s_1^N is then reshuffled by the permutation R_N to provide the vector $v_1^N = (s_1, s_3, \ldots, s_{N-1}, s_2, s_4, \ldots, s_N)$. The first half $(s_1, s_3, \ldots, s_{N-1})$ of v_1^N becomes the input to the first copy of $W_{N/2}$ and the second half $(s_2, s_4, \ldots, s_N)$ becomes the input to the second copy of $W_{N/2}$. This is repeated until we finally reach the N copies of W and transmit our transformed input vector.

From this definition, we can see that the blockwise transition probability is recursively transformed with the relation

$$W_{2N}(y_1^{2N}|u_1^{2N}) = W_N(y_1^N|u_{1,o}^{2N} \oplus u_{1,e}^{2N}) W_N(y_{N+1}^{2N}|u_{1,e}^{2N}). \tag{8}$$

We note that the transformation $u_1^N \mapsto x_1^N$ is *linear*. More specifically, notice that $W_N(y_1^N|u_1^N) = W^N(y_1^N|u_1^N G_N)$, where G_N is called the *generator matrix*. In channel polarization, we seek for a G_N which achieves polarization and at the same time gives low complexity encoding and decoding algorithms. Here we presented only one of those possible G_N matrices (transformations).

1.2.2 Channel Splitting

The *Channel Splitting* phase splits W_N back into a set of N binary-input coordinate channels $W_N^{(i)} : X \to Y^N \times X^{i-1}$ such that, if u_1^N is *a priori* uniform, $W_N^{(i)}$ is the effective channel seen by the ith input u_i, given both the actual channel output y_1^N and all the *previous* actual inputs u_1^{i-1}. Thus, the transition probabilities of the coordinate channels are defined by

$$\begin{aligned} W_N^{(i)}(y_1^N, u_1^{i-1}|u_i) &= \frac{W_N(y_1^N, u_1^i)}{P\{u_i\}} = \sum_{u_{i+1}^N \in X^{N-i}} \frac{W_N(y_1^N, u_1^i, u_{i+1}^N)}{P\{u_i\}} \\ &= \sum_{u_{i+1}^N \in X^{N-i}} \frac{W_N(y_1^N|u_1^N) P\{u_1^N\}}{P\{u_i\}} = \sum_{u_{i+1}^N \in X^{N-i}} \frac{W_N(y_1^N|u_1^N) 2^{-N}}{2^{-1}} \\ &= \sum_{u_{i+1}^N \in X^{N-i}} \frac{W_N(y_1^N|u_1^N)}{2^{N-1}} = \sum_{u_{i+1}^N \in X^{N-i}} \frac{1}{2^{N-1}} W_N(y_1^N|u_1^N). \end{aligned} \tag{9}$$

The effective channel $W_N^{(i)}$ will be used to estimate the input u_i. We will address the effect of this choice for decoding polar codes in Chapter 1.6. Henceforth, the coordinate channels $\{W_N^{(i)}\}$ will be called *bit-channels*, as they are used to transmit a single bit.

1.3 Transformation of Channel, Rate, and Reliability

We will now see how this blockwise transformation is broken into single-step channel transformations and then how rate and reliability transform alongside. Consider a binary-input channel $W : X \to Y$. A pair of binary-input channels $W' : X \to \tilde{Y}$ and $W'' : X \to \tilde{Y} \times X$ are obtained by a *single-step transformation* of two independent copies of W, denoted by $(W, W) \to (W', W'')$, iff there exists a one-to-one mapping $f : Y^2 \to \tilde{Y}$ such that

$$W'(f(y_1, y_2)|u_1) = \sum_{u_2' \in X} \frac{1}{2} W(y_1|u_1 \oplus u_2') W(y_2|u_2'), \tag{10}$$

$$W''(f(y_1, y_2), u_1|u_2) = \frac{1}{2} W(y_1|u_1 \oplus u_2) W(y_2|u_2). \tag{11}$$

In what follows, we will take f as the identity mapping. Hence, (10) and (11) simplify to

$$W'(y_1^2|u_1) = \sum_{u_2 \in X} \frac{1}{2} W(y_1|u_1 \oplus u_2) W(y_2|u_2), \tag{12}$$

$$W''(y_1^2, u_1|u_2) = \frac{1}{2} W(y_1|u_1 \oplus u_2) W(y_2|u_2). \tag{13}$$

Then, the general single-step channel transformations are of the form

$$\left(W_N^{(i)}, W_N^{(i)}\right) \mapsto \left(W_{2N}^{(2i-1)}, W_{2N}^{(2i)}\right) \tag{14}$$

and, more specifically,

$$\begin{aligned} W_{2N}^{(2i-1)}(y_1^{2N}, u_1^{2i-2}|u_{2i-1}) = \sum_{u_{2i} \in X} \frac{1}{2} W_N^{(i)}(y_1^N, u_{1,o}^{2i-2} \oplus u_{1,e}^{2i-2}|u_{2i-1} \oplus u_{2i}) \\ \cdot W_N^{(i)}(y_{N+1}^{2N}, u_{1,e}^{2i-2}|u_{2i}), \end{aligned} \tag{15}$$

$$W_{2N}^{(2i)}(y_1^{2N}, u_1^{2i-1}|u_{2i}) = \frac{1}{2} W_N^{(i)}(y_1^N, u_{1,o}^{2i-2} \oplus u_{1,e}^{2i-2}|u_{2i-1} \oplus u_{2i}) \cdot W_N^{(i)}(y_{N+1}^{2N}, u_{1,e}^{2i-2}|u_{2i}). \tag{16}$$

The proof of (15) and (16) is provided in the Appendix.

This result will allow us able to study the properties of the overall rate and reliability transformation. We will first study the transformation of rate of the local, single-step, channel transformation (14).

Proposition 2. *Consider $(W, W) \to (W', W'')$ for some set of binary-input channels. Then,*

$$I(W') + I(W'') = 2I(W), \tag{17}$$

$$I(W') \leq I(W''), \tag{18}$$

with equality in (18) iff $I(W)$ equals 0 or 1.

The proof of Proposition 2 is provided in the Appendix.

Equation (17) indicates that, under a single-step channel transformation, symmetric capacity is preserved. Equation (17) together with inequality (18) imply that $I(W') = I(W'') = I(W)$ iff $I(W)$ is either 1 or 0. In any other case, the single-step transformation *extremize* the symmetric capacity in the sense that

$$I(W') \leq I(W) \leq I(W''). \tag{19}$$

Next, we have the following results regarding the local-level transformation of reliability.

Proposition 3. *Consider* $(W, W) \to (W', W'')$ *for some binary-input channels. Then,*

$$Z(W'') = Z(W)^2, \tag{20}$$

$$Z(W') \leq 2Z(W) - Z(W)^2, \tag{21}$$

$$Z(W') \geq Z(W) \geq Z(W''), \tag{22}$$

with equality in (21) *iff* W *is a BEC.*

The proof of Proposition 3 is provided in the Appendix.

From (20), (21), and (22), we infer that $Z(W') = Z(W'') = Z(W)$ iff $Z(W)$ equals 0 or 1. This is equivalent to the implication we have from the transformation of rate. Also, we infer that reliability can only improve under a single-step transformation in the sense that

$$Z(W') + Z(W'') \leq 2Z(W). \tag{23}$$

At last, we have a result for the special case of the transformation of a BEC. Before proceeding to the statement, we will first define the *multi-output* BEC. Consider a symmetric binary-input channel $W'(y_1, \ldots, y_n|x) : X \to \underline{Y}$, where $\underline{Y} \in Y^n$. We denote with $\underline{Y}_i$, $1 \leq i \leq 2^n$, the possible output vectors of W'. We say that W' is BEC if its transition probabilities are of the form as in Figure 6, for arbitrary k, l. As it can be seen, the BEC defined in Chapter 1.1 is a special case of this definition for $k = 0$, $l = n = 1$. Now, we can state the anticipated result.

Proposition 4. *Consider the channel transformation* $(W, W) \to (W', W'')$. *If* W *is a BEC with some erasure probability* ϵ, *then the channels* W' *and* W'' *are BECs with erasure probabilities* $2\epsilon - \epsilon^2$ *and* ϵ^2, *respectively. Conversely, if either* W' *or* W'' *is a BEC, then* W *is BEC.*

The proof of Proposition 4 is provided in the Appendix.

Using recursively the results of Propositions 2 and 3, we derive the following proposition for the general case.

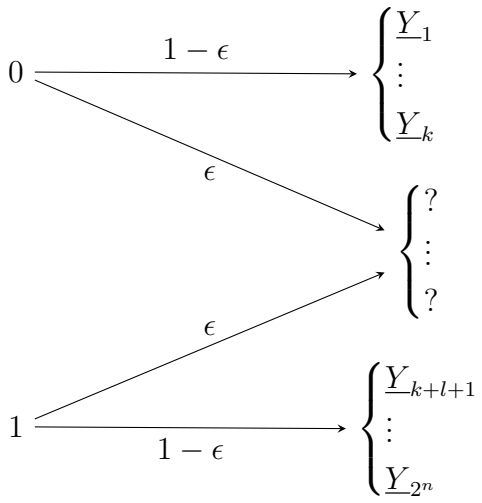

Figure 6: The multi-output binary erasure channel with erasure probability ε.

Proposition 5. *For any B-DMC W, $N = 2^n$, $n \geq 0$, $1 \leq i \leq N$, the transformation $(W_N^{(i)}, W_N^{(i)}) \to (W_{2N}^{(2i-1)}, W_{2N}^{(2i)})$ is rate-preserving and reliability-improving in the sense that*

$$I\left(W_{2N}^{(2i-1)}\right) + I\left(W_{2N}^{(2i)}\right) = 2I\left(W_N^{(i)}\right), \tag{24}$$

$$Z\left(W_{2N}^{(2i-1)}\right) + Z\left(W_{2N}^{(2i)}\right) \leq 2Z\left(W_N^{(i)}\right), \tag{25}$$

with equality in (25) iff W is a BEC. Channel splitting extremizes the rate and reliability in the sense that

$$I\left(W_{2N}^{(2i-1)}\right) \leq I\left(W_N^{(i)}\right) \leq I\left(W_{2N}^{(2i)}\right), \tag{26}$$

$$Z\left(W_{2N}^{(2i-1)}\right) \geq Z\left(W_N^{(i)}\right) \geq Z\left(W_{2N}^{(2i)}\right), \tag{27}$$

with equality in (26) and (27) iff $I(W)$ equals 0 or 1. The reliability terms further satisfy

$$Z\left(W_{2N}^{(2i-1)}\right) \leq 2Z\left(W_N^{(i)}\right) - Z\left(W_N^{(i)}\right)^2, \tag{28}$$

$$Z\left(W_{2N}^{(2i)}\right) = Z\left(W_N^{(i)}\right)^2, \tag{29}$$

with equality in (28) iff W is BEC. The cumulative rate and reliability satisfy

$$\sum_{i=1}^{N} I(W_N^{(i)}) = NI(W), \tag{30}$$

$$\sum_{i=1}^{N} Z(W_N^{(i)}) \leq NZ(W), \tag{31}$$

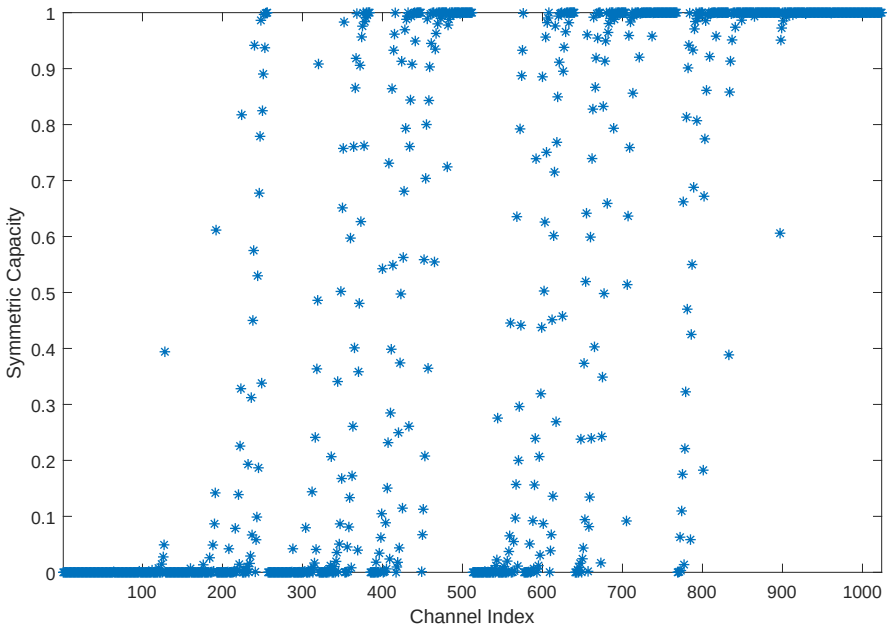

Figure 7: N = 1024 polarized copies of BEC($\epsilon = 0.5$).

with equality in (31) *iff* W *is a BEC.*

For the special case that W is a BEC with an erasure probability ϵ, the parameters $\{Z(W_N^{(i)})\}$ and $\{I(W_N^{(i)})\}$ can be computed through the recursions

$$\begin{aligned} Z\left(W_N^{(2i-1)}\right) &= 2Z\left(W_{N/2}^{(i)}\right) - Z\left(W_{N/2}^{(i)}\right)^2, \\ Z\left(W_N^{(2i)}\right) &= Z\left(W_{N/2}^{(i)}\right)^2, \end{aligned} \tag{32}$$

$$\begin{aligned} I\left(W_N^{(2i-1)}\right) &= I\left(W_{N/2}^{(i)}\right)^2, \\ I\left(W_N^{(2i)}\right) &= 2I\left(W_{N/2}^{(i)}\right) - I\left(W_{N/2}^{(i)}\right)^2, \end{aligned} \tag{33}$$

with $Z(W_1^{(1)}) = \epsilon$. The parameter $Z(W_N^{(i)})$ equals the erasure probability of the channel $W_N^{(i)}$. The recursion (33) follows from (32) by the fact that $I(W_N^{(i)}) = 1 - Z(W_N^{(i)})$ for $W_N^{(i)}$ a BEC (6). We use the above recursions to illustrate the polarization effect *per channel* in Fig. 7.

1.4 Relation between the Construction and the Index of a bit-channel

We showed that channel polarization is broken down to single-step channel transformations using the relations (15) and (16), which we repeat below.

$$W_{2N}^{(2i-1)}(y_1^{2N}, u_1^{2i-2}|u_{2i-1}) = \sum_{u_{2i}} \frac{1}{2} W_N^{(i)}(y_1^N, u_{1,o}^{2i-2} \oplus u_{1,e}^{2i-2}|u_{2i-1} \oplus u_{2i}) \cdot W_N^{(i)}(y_{N+1}^{2N}, u_{1,e}^{2i-2}|u_{2i}), \tag{15}$$

$$W_{2N}^{(2i)}(y_1^{2N}, u_1^{2i-1}|u_{2i}) = \frac{1}{2} W_N^{(i)}(y_1^N, u_{1,o}^{2i-2} \oplus u_{1,e}^{2i-2}|u_{2i-1} \oplus u_{2i}) \cdot W_N^{(i)}(y_{N+1}^{2N}, u_{1,e}^{2i-2}|u_{2i}). \tag{16}$$

We see that the path of a transformed channel inside the above recursions is directly related with its index. By making the convention that bit-channel indexing ranges from 1 through N, but their correspondent binary representation ranges from all zeros to all ones (i.e. 1 is represented by '000...0', 2 is represented by '000...1' etc.), when two copies of a channel with index $(b_1, b_2, \ldots, b_n)_2$ are transformed, we get two new channels, one even-indexed and one odd-indexed with indices $(b_1, b_2, \ldots, b_n, 0)_2$ and $(b_1, b_2, \ldots, b_n, 1)_2$, respectively. It follows that we can construct a bit-channel recursively simply by following the binary representation of its index (from the MSB to the LSB), using either (15) or (16) when the next bit is 0 or 1.

1.5 Channel Polarization: Main results

We are now ready to prove the main results of channel polarization.

Theorem 1. *For any B-DMC W, the channels $\left\{W_N^{(i)}\right\}$ polarize in the sense that, for any fixed $\delta \in (0,1)$, as N goes to infinity through powers of two, the fraction of indices $i \in \{1, \ldots, N\}$ for which $I(W_N^{(i)}) \in (1-\delta, 1]$ goes to $I(W)$ and the fraction for which $I(W_n^{(i)}) \in [0, \delta)$ goes to $1 - I(W)$.*

Before we proceed to the proof, we will first construct the framework upon which we will work. We define a binary tree, which represents the channel transformation procedure (14). The tree is illustrated in Fig. 8. Notice that channel $W_{2^n}^{(i)}$ is located at the nth level of the tree at node number i counting from top. We index each channel-node with bit sequences. The root node is indexed with the null sequence. The upper node at level 1 is indexed with 0 and the lower node with 1. Given a node at level n with index $b_1 b_2 \ldots b_n$, the upper node emanating from it is indexed with $b_1 b_2 \ldots b_n 0$ and the lower with $b_1 b_2 \ldots b_n 1$. We denote the channel $W_{2^n}^{(i)}$ located at node $b_1 b_2 \ldots b_n$, as $W_{b_1 \ldots b_n}$.

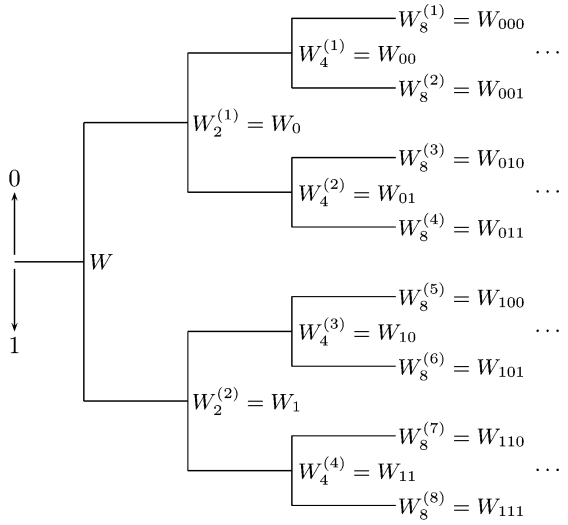

Figure 8: The tree process for the recursive channel construction.

Upon this tree, we define a uniform random tree process $\{K_n : n \geq 0\}$, where $P(K_n = W_{b_1 \ldots b_n}) = 1/2^n$, for every sequence $b_1 \cdots b_n$. For the initial step, we define $K_0 = W$. To keep track of the rate and reliability parameters of the random sequence of channels K_n, we define the random processes $I_n = I(K_n)$ and $Z_n = Z(K_n)$. Clearly, $I_0 = I(W)$ and $Z_0 = Z(W)$.

More precisely, consider the probability space $(\Omega, \Im, P)$. The sample space Ω is the space of all binary sequences $(b_1, b_2, \ldots) \in \{0,1\}^\infty$, i.e. all the probable paths on the infinite random tree process. To reach the end of these paths we need an infinite amount of single-step channel transformations. The set $\Im$ is generated by the binary sequences $S(b_1, \ldots, b_n) \triangleq \{\omega \in \Omega : \omega_1 = b_1, \ldots, \omega_n = b_n\}$, $n \geq 1$, where $b_1, \ldots, b_n \in \{0,1\}$. That is, the *cylinder set* $S(b_1, \ldots, b_n)$ includes all the paths on the infinite random tree process which start with the sequence $b_1, \ldots, b_n$, and $\Im$ includes all of those cylinder sets. The function $P(\cdot)$ is the probability measure defined on $\Im$, such that $P(S(b_1, \ldots, b_n)) = 1/2^n$. Notice that $S(b_1, \ldots, b_n) = S(b_1, \ldots, b_n, 0) \cup S(b_1, \ldots, b_n, 1)$. For each $n \geq 0$, we define $\Im_n$ as the set generated by the cylinder sets $S(b_1, \ldots, b_i)$, $1 \leq i \leq n$, $b_1, \ldots, b_i \in \{0,1\}$. We define $\Im_0$ as the set consisting of the empty set and Ω only. Clearly, $\Im_0 \subset \Im_1 \subset \ldots \subset \Im$. Now we are ready to prove Theorem 1.

Proof of Theorem 1: We will employ the following two propositions:

Proposition 6. *The sequence* $\{I_n, \Im_n; n \geq 0\}$ *is a martingale:*

$$\Im_n \subset \Im_{n+1} \text{ and } I_n \text{ is } \Im_n\text{-measurable} \tag{34}$$

$$E[|I_n|] < \infty \tag{35}$$

$$I_n = E[I_{n+1}|\Im_n]. \tag{36}$$

Furthermore, the sequence $\{I_n; n \geq 0\}$ *converges almost everywhere to a random variable* I_∞ *such that* $E[I_\infty] = I_0$.

Proof: Condition (34) is true by construction and (35) by the fact that $0 \leq I_n \leq 1$. To prove (36) we use (24) to write

$$E[I_{n+1}|S(b_1, \ldots, b_n)] = \frac{1}{2}I(W_{b_1 \ldots b_n 0}) + \frac{1}{2}I(W_{b_1 \ldots b_n 1}) = I(W_{b_1 \ldots b_n}) = I_n. \tag{37}$$

Since I_n is bounded, $I(W_{b_1 \ldots b_n})$ is a uniformly integrable martingale. We use [3, Theorems 9.4.5, 9.4.6] to state that I_n converges almost everywhere to a random variable I_∞ and $E[I_n] \longrightarrow E[I_\infty]$. We use (30) to derive that $E[I_n] = I_0$ for any $n \geq 0$. Therefore, $E[I_\infty] = I_0$. □

Proposition 7. *The sequence* $\{Z_n, \Im_n; n \geq 0\}$ *is a supermartingale:*

$$\Im_n \subset \Im_{n+1} \text{ and } Z_n \text{ is } \Im_n\text{-measurable} \tag{38}$$

$$E[|Z_n|] < \infty \tag{39}$$

$$Z_n \geq E[Z_{n+1}|\Im_n]. \tag{40}$$

Furthermore, the sequence $\{Z_n; n \geq 0\}$ *converges almost everywhere to a random variable* Z_∞ *which takes values almost everywhere in* $\{0, 1\}$.

Proof: Condition (38) is satisfied by construction. Condition (39) is satisfied by the fact that $0 \leq Z_n \leq 1$. To prove, (40) we use (25) to write

$$E[Z_{n+1}|S(b_1, \ldots, b_n)] = \frac{1}{2}Z(W_{b_1 \ldots b_n 0}) + \frac{1}{2}Z(W_{b_1 \ldots b_n 1}) \leq Z(W_{b_1 \ldots b_n}) = Z_n. \tag{41}$$

Since Z_n is bounded, $Z(W_{b_1 \ldots b_n})$ is a uniformly integrable martingale. We use [3, Theorem 9.4.5] to state that Z_n converges almost everywhere to a random variable Z_∞, such that $E[|Z_n - Z_\infty|] \longrightarrow 0$. It follows that $E[|Z_{n+1} - Z_n|] \longrightarrow 0$. But, by (28) and (29) we derive that $Z_{n+1} = {Z_n}^2$ with probability 1/2 $Z_{n+1} > {Z_n}^2$ with probability 1/2 (these are the cases where we choose the lower or upper subtree when we make a random step from a node). Hence, $E[|Z_{n+1} - Z_n|] \geq (1/2)E[{Z_n}^2 - Z_n] \geq 0$. Thus, $E[Z_n(1 - Z_n)] \longrightarrow 0$, which implies $E[Z_\infty(1 - Z_\infty)] \longrightarrow 0$. But $Z_\infty \in [0, 1]$ and so $(1 - Z_\infty) \in [0, 1]$ as well. Hence, $(Z_\infty(1 - Z_\infty)) \in [0, 1]$. This implies that $(Z_\infty(1 - Z_\infty))$ equals 0 almost everywhere

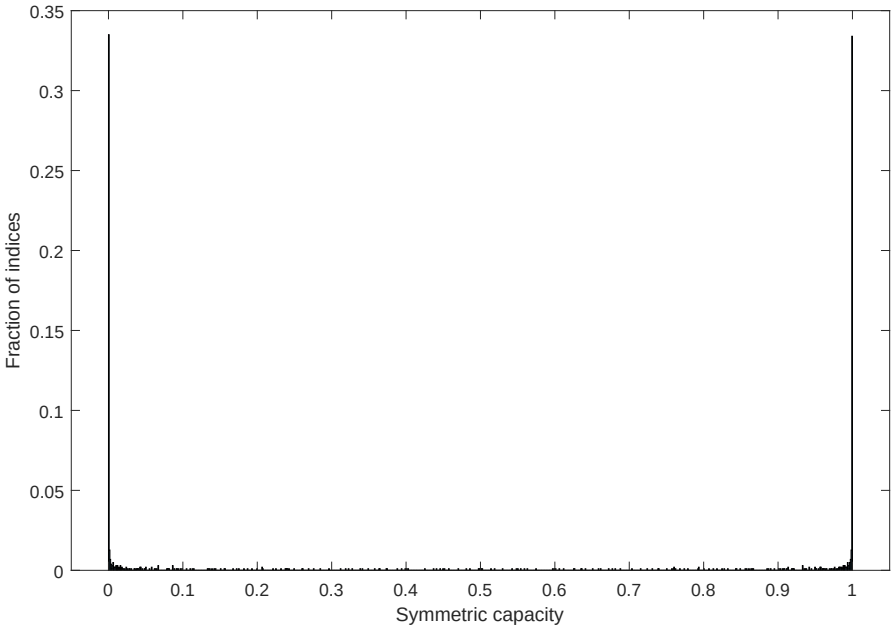

Figure 9: Polarization of BEC ($\epsilon = 0.5$) at the 1024th level.

or, equivalently, that Z_∞ equals 0 or 1 almost everywhere. This completes the proof of Proposition 7. □

The fact that Z_∞ equals 0 or 1 almost everywhere, combined with Corollary 1, implies that $I_\infty = 1 - Z_\infty$ almost everywhere, and hence I_∞ equals 1 or 0 almost everywhere.

To complete the proof of Theorem 1, we notice that the sequence $\{I_\infty^{(i)}\}$ is an infinite Bernoulli process. Hence $P(I_\infty^{(i)} = 1) = E[I_\infty] = I_0$. By Borel's Law of Large Numbers, we have that the fraction of indices i for which $I_\infty^{(i)} = 1$ is equal to $P(I_\infty^{(i)} = 1)$, namely I_0, and the fraction for which $I_\infty^{(i)} = 0$ is equal to $1 - I_0$. This concludes the proof of Theorem 1. □

Fig. 9 illustrates the validity of Theorem 1. We see that, for a BEC with $\epsilon = 0.5$, at the 1024th level of the tree process, almost 70% of I_n take values in $\{0 + \delta, 1 - \delta\}$ for a small δ.

Theorem 2. *For any B-DMC W with $I(W) > 0$ and any fixed $R < I(W)$, there exists a sequence of sets $A_N \subset \{1, \ldots, N\}$, $N \in \{1, 2, \ldots, 2^n, \ldots\}$, such that $A_N \geq NR$ and $Z(W_N^{(i)}) \leq O(N^{-5/4})$ for all $i \in A_N$.*

The proof of Theorem 2 is provided in the Appendix.

We stated the polarization result in Theorem 2 in terms of $\{Z(W_N^{(i)})\}$. A rate of polarization result in terms of $\{I(W_N^{(i)})\}$ can be obtained from Theorem 2 with the help of Proposition 1.

1.6 Polar Coding

In the presence of channel polarization, coding becomes trivial: We send data only through the bit-channels for which $Z(W_N^{(i)})$ is near 0. We call this coding method *polar coding*.

Individual codes will be identified by a parameter vector (N, K, A, u_{A^c}), where $N = 2^n$ is the number of the available bit-channels, K is the code dimension and specifies the size of A, where A is a fixed subset of the bit-channels which will be used to send information and $u_{A^c \in X^{N-K}}$ is a fixed vector that is sent over the subset A^c, which is the complement of A over all N bit-channels. The number N is the block length and the ratio $K/N = R$ is the code rate. We will refer to A as the *information set*, whereas A^c will be referred to as the *frozen set*. Accordingly, $u_A \in X^K$ will be referred to as the *information vector* and u_{A^c} as the *frozen vector*.

1.6.1 Successive Cancellation Decoding

Consider a code with parameter (N, K, A, u_{A^c}). Let u_1^N be encoded into a codeword x_1^N, let x_1^N be sent over the channel W^N, and let a channel output y_1^N be received. The decoder's task is to generate an estimate $\hat{u}_1^N$ of u_1^N, given knowledge of A, u_{A^c} and y_1^N.

The successive cancellation (SC) decoder generates its decision vector by computing

$$\hat{u}_1^N \triangleq \begin{cases} u_i, & \text{if } i \in A^c, \\ h_i(y_1^N, \hat{u}_1^{i-1}), & \text{if } i \in A, \end{cases} \tag{42}$$

in the order i from 1 to N, where $h_i : Y^N \times X^{i-1} \to X$, $i \in A$, are *decision functions* defined as

$$h_i(y_1^N, \hat{u}_1^{i-1}) \triangleq \begin{cases} 0, & \text{if } \frac{W_N^{(i)}(y_1^N, \hat{u}_1^{i-1}|0)}{W_N^{(i)}(y_1^N, \hat{u}_1^{i-1}|1)} \geq 0, \\ 1, & \text{otherwise.} \end{cases} \tag{43}$$

We say that a decoding *block error* occured if $\hat{u}_1^N \neq u_1^N$ or equivalently, if $\hat{u}_A \neq u_A$.

1.6.2 Probability of error

The notation $P_e(N, K, A, u_{A^c})$ will denote the probability of block error for a (N, K, A, u_{A^c}) code, assuming that each information vector $u_A \in X^K$ is sent with probability 2^{-K} and decoding is done by the above SC decoder. More precisely,

$$P_e(N, K, A, u_{A^c}) \triangleq \sum_{u_A \in X^K} \frac{1}{2^K} \sum_{y_1^N \in Y^N : \hat{u}_1^N(y_1^N) \neq u_1^N} W_N(y_1^N | u_1^N). \tag{44}$$

The average of $P_e(N, K, A, u_{A^c})$ over all choices for $u_{A^c} \in X^{N-K}$ will be denoted as $P_e(N, K, A)$.

Proposition 8. *For any B-DMC W and any choice of the parameters (N, K, A)*

$$P_e(N, K, A) \leq \sum_{i \in A} Z\left(W_N^{(i)}\right). \tag{45}$$

Proof: We may express the block error event as $\varepsilon = \cup_{i \in A} B_i$, where B_i is the event that the first decision error in SC decoding occurs at stage i. We notice that

$$\begin{aligned} B_i &\triangleq \{(u_1^N, y_1^N) \in X^N \times Y^N : u_1^{i-1} = \hat{U}_1^{i-1}(u_1^N, y_1^N),\ u_i \neq h_i\left(y_1^N, \hat{U}_1^{i-1}(u_1^N, y_1^N)\right)\} \\ &= \{(u_1^N, y_1^N) \in X^N \times Y^N : u_1^{i-1} = \hat{U}_1^{i-1}(u_1^N, y_1^N),\ u_i \neq h_i(y_1^N, u_1^{i-1})\} \\ &\subset \{(u_1^N, y_1^N) \in X^N \times Y^N : u_i \neq h_i(y_1^N, u_1^{i-1}\} \subset \varepsilon_i, \end{aligned} \tag{46}$$

where ε_i is the error event when the i-th coordinate channel is used only once to transmit a bit. Thus, we have

$$P(\varepsilon) \leq \sum_{i \in A} P(\varepsilon_i). \tag{47}$$

Now, using (3) we conclude that

$$P(\varepsilon) \leq \sum_{i \in A} Z\left(W_N^{(i)}\right). \tag{48}$$

□

Proposition 8 leads to the idea behind the definition of polar codes.

1.6.3 Polar Codes

Given a B-DMC W, a code with parameter (N, K, A, u_{A^c}) will be called a *polar code* for W if the information set A is chosen as a K-element subset of $\{1, ..., N\}$ such that

$Z(W_N^{(i)}) \leq Z(W_N^{(j)})$ for all $i \in A$, $j \in A^c$. That is, the information set A is chosen as to minimize the RHS of (48).

Polar codes are *channel-specific* codes: a polar code for one channel may not be a polar code for another.

The choice of u_{A^c} is negligible for the performance of polar codes. In fact, below we state that for symmetric channels, any choice for u_{A^c} is as good as any other.

We complement Theorem 2 with the following results, proved in [1, Theorems 3-4].

Theorem 3. *For any given B-DMC W and fixed $R < I(W)$, block error probability for polar coding under successive cancellation decoding satisfies*

$$P_e(N, R) = O(N^{-\frac{1}{4}}). \tag{49}$$

Theorem 4. *For any symmetric B-DMC W and any fixed $R < I(W)$, consider any sequence of (N, K, A, u_{A^c}) codes with N increasing to infinity, $K = \lfloor NR \rfloor$, A chosen in accordance with the polar coding rule for W, and u_{A^c} fixed arbitrarily. The block error probability under successive cancellation decoding satisfies*

$$P_e(N, K, A, u_{A^c}) = O(N^{-\frac{1}{4}}). \tag{50}$$

Note: In more recent works the bounds in Theorems 2,3 & 4 have been strengthened. More precisely, in [4] it is shown that for any binary-input discrete memoryless channel W with symmetric capacity $I(W)$ and any rate $R < I(W)$, the probability of block decoding error for polar coding under successive cancellation decoding satisfies

$$P_e = O(2^{-N^\beta}) \tag{51}$$

for any $\beta < \frac{1}{2}$ when the block-length N is large enough.

In [5] a rate-dependent bound is derived: For any B-MC W with $I(W) > 0$ and fixed $R < I(W)$, the best achievable block error probability satisfies

$$P_e(N, R) = o(2^{-2^{(n+t\sqrt{n})/2}}), \tag{52}$$

for any t satisfying $t < Q^{-1}(R/I(W))$, where $Q(x) = \int_\chi^\infty e^{-u^2/2} du/\sqrt{2\pi}$.

We now return to the observation we first made in Chapter 1.2.2. While the synthesized channel $W_N^{(i)}$ has in its output vector the *actual* inputs u_1^{i-1}, when estimating the input u_i, the SC decoder knows only their *estimations* $\hat{u}_1^{i-1}$. But, following the results of channel polarization and the definition of polar codes, we know that the estimations are correct with probability which tends to 1, because they are either in the frozen vector, and thus already known to the decoder, or they are sent over a bit-channel with Bhattacharyya parameter close to 0.

1.6.4 Complexity of Encoding and Decoding Polar Codes

It is proved and well known that both encoding and successive cancellation decoding of polar codes have complexity $O(N \log N)$, where N is the block length. For the part of the encoding, the main idea is to notice the recursive nature of channel combining, which is given schematically in Figure 5. As for the decoding part, applying (15) and (16) in (43), we take a binary recursive relation as well. The above are stated and proved analytically in [1, VII, VIII].

2 Construction of Polar Codes

Having in our hands a low complexity encoding and decoding channel code, it is of great interest the efficient construction of such a code. That is, the determination of the information set A according to the rule described in the definition of polar codes. The construction of polar codes poses many challenges, like having the ability to adapt rapidly in channel, rate and block length variations. As we already mentioned, polar codes are channel-specific codes, which means that when the physical channel changes dynamically, as it happens with mobile communications, we have to re-construct a new information set. Polar codes are chosen to be used in 5G NR, which strengthens the demand for a fast and adaptive construction algorithm.

In what follows, we will restrict our results in BSCs, but we can generalize the procedure for any other BMS channel. We note that only for the BEC there is a fast, memory-efficient and optimal construction algorithm, in terms of calculating efficiently the Bhattacharyya parameters of the bit-channels, using (32). For any other channel, a *robust* calculation of the Bhattacharyya parameters is still either cumputationally highly demanding or impractical due to memory requirements. Here we will combine the approximation method proposed in [7], which scales linearly with the block length, with the partial orders of the bit-channels proposed in [8].

2.1 The Construction Problem

Given an arbitrary BMS channel W and a block length N, while trying to construct the information set A, one has to answer to either of the following two questions:

1. *Fixed-Rate Construction:* What rate R am I trying to achieve?

2. *Fixed-Performance Construction:* What block error rate P_e am I trying to achieve?

Those two questions are of course equivalent in the following sense: a certain rate R achieves a certain block error rate P_e for a given BMS channel W and block length N, and vice versa. Usually, one wants to transmit information with a certain upper bound on the bit error rate. However, we will concentrate on the fixed-rate construction problem. This is because, as we'll see, it can be solved faster, and on the other hand a fixed-performance problem can easily be converted to a fixed-rate using a bijection vector.

2.1.1 Solving strategy of the Fixed-Rate Construction Problem

More precisely, the problem we are trying to solve is the following: Given an arbitrary BMS channel W, a block length N and a code rate $R = K/N$, which K bit-channels minimize the RHS of (48)? In other words, which K out of the N bit-channels have the lowest Bhattacharyya parameter - or - are *more reliable* than the rest $N - K$?

Now we will state our solving strategy, which is inspired by [6]. Given an arbitrary BMS channel W, a block length N and a rate $R = K/N$, suppose that someone gives us the following information: Some bit-channels W_i, $i \in \{0, \dots, N-1\}$ are more reliable than at least $N - K$ bit-channels and some other bit-channels W_j, $j \in \{0, \dots, N-1\}$ are less reliable than at least K bit-channels. We denote the former of those subsets I and the latter F. Obviously, $I \bigcap F = \{\emptyset\}$. Then, $I \in A$ and $F \in A^c$. We denote the rest (undetermined) bit-channels as $U = \{W_k,\ k \in \{0, \dots, N-1\} : W_k \notin I \bigcup F\}$. Then, in order to solve the fixed-rate construction problem, it suffices to totally order the reliability of the bit-channels that are in U: The most reliable of them will complete A and the least reliable will complete A^c.

The above strategy proposes two complementary ways to construct the information set A. The first is to use the pair-wise reliability relation between the bit-channels, as much as is known to us such a relation, and conclude if some of them are in I or F and the second is to decide -in any way possible- which of the rest are explicitly the most reliable. The performance gains by solving the fixed-rate problem instead of the fixed-performance comes from noticing that we don't need to precisely calculate the Bhattacharyya parameters of the subset U. On the contrary, this would be mandatory if we tried to solve the fixed-performance problem.

In the following sections, we lay out the theoretical tools which we use for accomplishing this strategy efficiently.

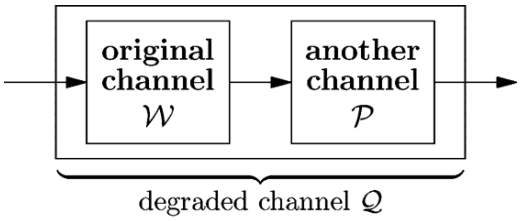

Figure 10: Q is degraded with respect to W.

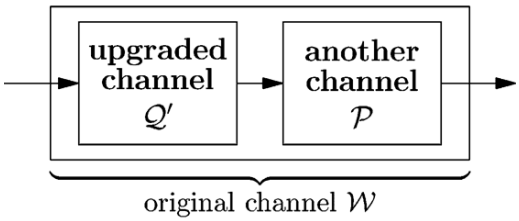

Figure 11: Q' is upgraded with respect to W.

2.2 Stochastically Degraded and Upgraded Channels

Obviously, a key to the proposed construction method is to define a reliability relation between the bit-channels. As in [7], we define the (stochastically) *degraded with respect to* and *upgraded with respect to* relations between two channels.

Let Q and W be two BMS channels $Q : X \to Z$, $W : X \to Y$. We say that Q is *stochastically degraded with respect to* W, denoted as $Q \preccurlyeq W$, if there exists a channel $P : Y \to Z$ such that

$$Q(z|x) = \sum_{y \in Y} W(y|x)P(z|y) \tag{53}$$

for all $z \in Z$ and $x \in X$.

Let Q' and W be two BMS channels $Q' : X \to Z'$, $W : X \to Y$. We say that Q' is *stochastically upgraded with respect to* W, denoted as $Q' \succcurlyeq W$, if there exists a channel $P : Z' \to Y$ such that,

$$W(y|x) = \sum_{z' \in Z'} Q'(z'|x)P(y|z') \tag{54}$$

for all $z' \in Z'$ and $x \in X$.

In other words, Q is said to be degraded with respect to W if there exists another channel P which if it intervenes after W's output, it produces Q's output, and Q' is said to be upgraded with respect to W if it can be degraded to W. Figures 11 and 10 illustrate the defined relations. Obviously,

$$Q' \succcurlyeq W \text{ iff } W \preccurlyeq Q'. \tag{55}$$

It can be shown that $\preccurlyeq$ and $\succcurlyeq$ are reflexive and transitive relations and thus,

$$W \preccurlyeq W \text{ and } W \succcurlyeq W \text{ and also,} \tag{56}$$

$$\text{if } W \preccurlyeq W' \text{ and } W' \preccurlyeq W'', \text{ then } W \preccurlyeq W''. \tag{57}$$

If two channels W and W' are both degraded with respect to each other, then we say that W and W' are *equivalent*, and denote this by $W \equiv W'$. By (55), (56) and (57) we get

that $\equiv$ is an equivalence relation and so,

$$W \equiv W, \tag{58}$$

$$W \equiv W' \text{ iff } W' \equiv W. \tag{59}$$

Now we will see the effect of $\preccurlyeq$ and $\succcurlyeq$ on three channel parameters of interest, namely, the probability of error under ML-decoding and uniform input, $P_e(\cdot)$, the Bhattacharyya parameter $Z(\cdot)$ and the symmetric capacity $I(\cdot)$. The next lemma is proved in [7].

Lemma 1. *Let $W : X \to Y$ be a BMS channel and suppose that $Q : X \to Z$ is degraded with respect to W. Then,*

$$P_e(Q) \geq P_e(W), \tag{60}$$

$$Z(Q) \geq Z(W), \tag{61}$$

$$I(Q) \leq I(W). \tag{62}$$

Because of (55), if we replace "degraded" with "upgraded", the inequalities are reversed. Therefore, if $W \equiv Q$, then the inequalities become equalities.

2.3 Universal partial orders of the bit-channels

In [8] it is proved that partial orders (PO) [9, Ch. 1.1] of reliability exist for the bit-channels of polarized symmetric channels with binary inputs. Below we restate the definitions of those POs and the main theorems from [8] without providing their proofs.

Let $i, j \in \{0, \cdots, N-1\}$ be the indices of the bit-channels $W_N^{(i)}$ and $W_N^{(j)}$ according to the transformation relations in (15) and (16). Let those indices have binary representations $(i_{n-1}, i_{n-2}, \cdots, i_0)_2$ and $(j_{n-1}, j_{n-2}, \cdots, j_0)_2$ respectively.

Theorem 5. *If for all $k \in \{0, \cdots, n-1\}$ we have $j_k = 1 \Rightarrow i_k = 1$, then $W_N^{(j)} \preccurlyeq W_N^{(i)}$.*

Examples: $W_N^{(010)} \preccurlyeq W_N^{(011)}$, $W_N^{(1001)} \preccurlyeq W_N^{(1011)}$.

Definition 1. *We write $j \nearrow i$ if there exist $l, l' \in \{0, \cdots, n-1\}$ with $l < l'$ such that*

1. $j_l = 1$ *and* $j_{l'} = 0$.
2. $i_l = 0$ *and* $i_{l'} = 1$.
3. *For all* $k \in \{0, \cdots, n-1\} \setminus \{l, l'\} : j_k = i_k$.

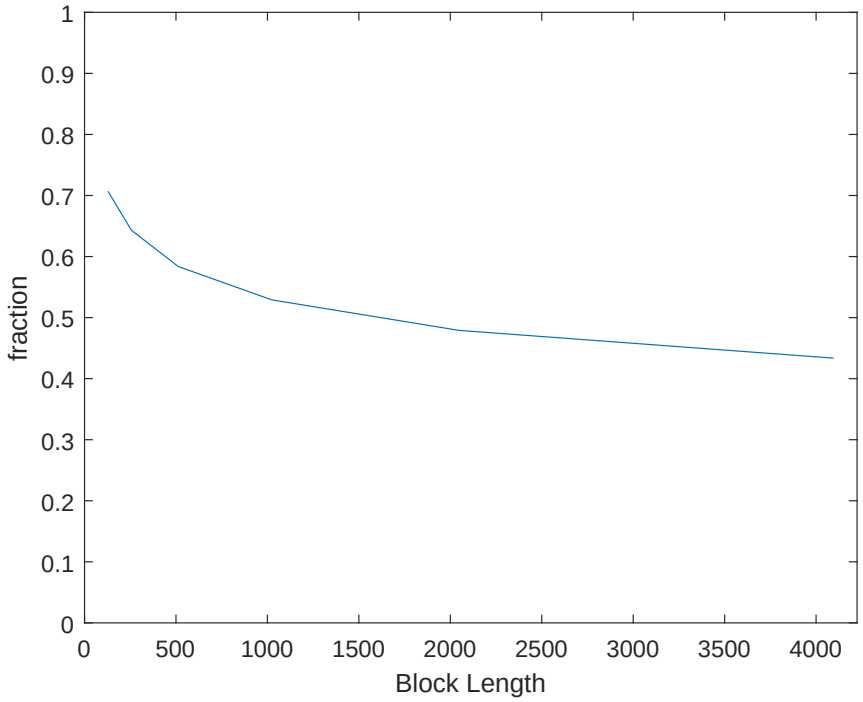

Figure 12: Pre-determined pairwise reliability orderings of bit-channels.

Essentially, $j \nearrow i$ if and only if we can obtain j by switching a more significant 1 with a less significant 0 in i.

Examples: $(01101) \nearrow (10101)$, $(0111) \nearrow (1110)$.

Theorem 6. *If $j \nearrow i$ then $W_N^{(j)} \preccurlyeq W_N^{(i)}$.*

The next proposition shows that we can combine the above POs, by using them explicitly on partitions of the binary representations of the indices. It is stated in [8], but we re-state and re-prove it here more clearly.

Proposition 9. *Let two bit-channels $W_N^{(i)}$ and $W_N^{(j)}$ with indices with binary representations $i = (i_n, i_{n-1}, \ldots, i_1)_2$ and $j = (j_n, j_{n-1}, \ldots, j_1)_2$ respectively. If the available partial orders can be applied explicitly on the indices' partitions $(i_{n-k_1}^n, j_{n-k_1}^n), (i_{n-k_1-1-k_2}^{n-k_1-1}, j_{n-k_1-1-k_2}^{n-k_1-1})$, $(i_{n-k_1-k_2-2-k_3}^{n-k_1-k_2-2}, j_{n-k_1-k_2-2-k_3}^{n-k_1-k_2-2}), \ldots,$ $(i_1^{n-(\sum_{i=1}^{r-1} k_i)-(r-1)}, j_1^{n-(\sum_{i=1}^{r-1} k_i)-(r-1)})$, where $(\sum_{i=1}^{r-1} k_i) - (r-1) < n$, and all give the same order direction, then the overall binary representations of those bit-channels follow the same reliability order as their partitioned counterparts.*

Proof: For shorthand we will notate the channels with indices the partitions

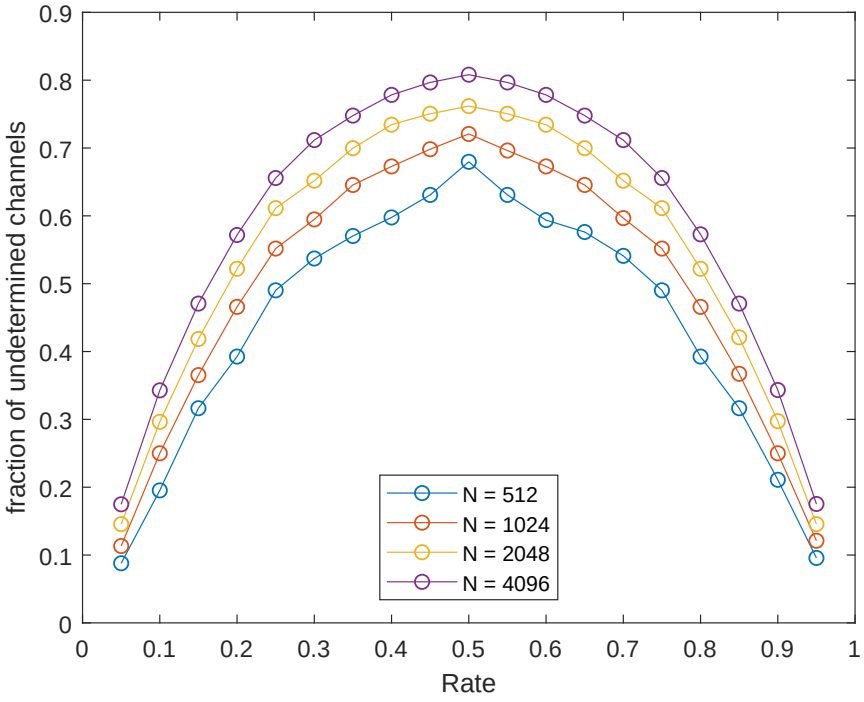

Figure 13: Percentage of the undetermined among the N bit-channels.

$i_{n-k_1}^{n}$ as i_{k_1}, $i_{n-k_1-1-k_2}^{n-k_1-1}$ as i_{k_2} and so on. Consider that from the above POs we have $(j_{k_1} \succcurlyeq i_{k_1})$, $(j_{k_2} \succcurlyeq i_{k_2})$, $(j_{k_3} \succcurlyeq i_{k_3})$, $\ldots$, $(j_1 \succcurlyeq i_1)$. Then, using the given partial orders we have $j = (j_{k_1} j_{k_2} \cdots j_{k_{r-1}} j_{k_r}) \succcurlyeq (j_{k_1} j_{k_2} \cdots j_{k_{r-1}} i_{k_r}) \succcurlyeq (j_{k_1} j_{k_2} \cdots i_{k_{r-1}} i_{k_r}) \succcurlyeq \cdots \succcurlyeq (i_{k_1} i_{k_2} \cdots i_{k_{r-1}} i_r) = i$. The same holds even when we switch "$\succcurlyeq$" with "$\preccurlyeq$". □

Example: The pair $W_N^{(0101001)_2}, W_N^{(0111100)_2}$ cannot be ordered from Theorems 5 and 6. But from Theorem 5 we know that $W_N^{(0101)_2} \preccurlyeq W_N^{(0111)_2}$, and from Theorem 6 that $W_N^{(001)_2} \preccurlyeq W_N^{(100)_2}$ Because the order direction is the same for both partitions, we derive that $W_N^{(0101001)_2} \preccurlyeq W_N^{(0111100)_2}$.

Theorems 5-6 and Proposition 9 give us a powerful means for constructing polar codes. Although polar codes are channel-specific, they claim a universal relation between the reliability of some of their bit-channels, which can be determined only by their indices. Note that this is a natural consequence derived from the transformations (15) and (16).

Figure 12 shows the fraction of the pair-wise reliability orderings from the overall $\binom{L}{2}$ bit-channel pairs, which can be obtained by combining the above POs according to Proposition 9. Figure 13 shows the efficiency we gain by utilizing the given partial orders.

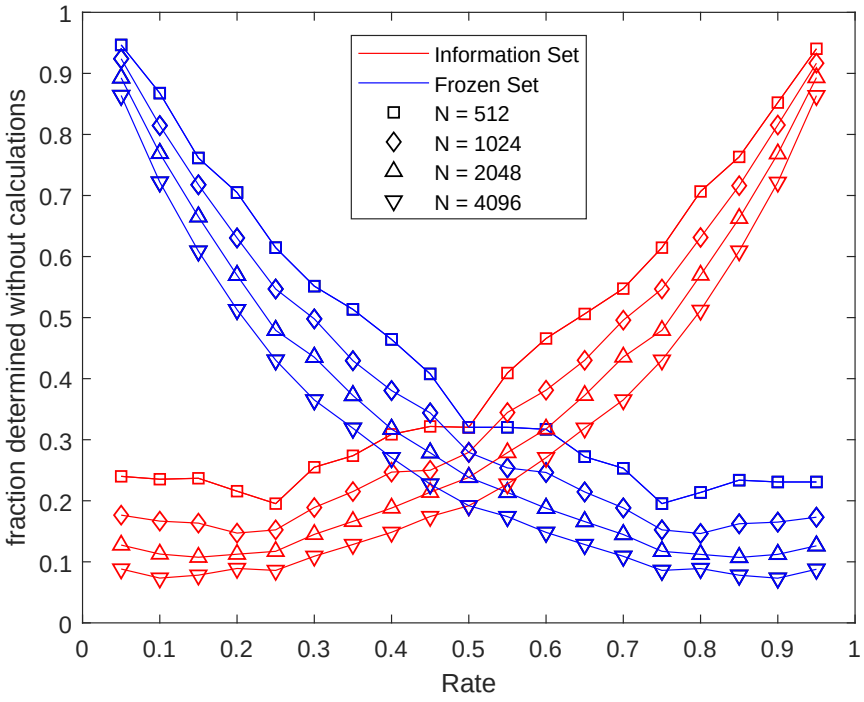

Figure 14: Fraction of I and F in the Information and Frozen sets.

Figure 14 shows the percentage of the information and the frozen vectors we can fill by employing the above POs in the proposed strategy. We notice that for low code rates, we can fill a significant portion of the frozen vector and for high rates a significant portion of the information vector.

Theorems 5 and 6 hold true regardless of the physical channel, so they are extremely useful for a method *adaptive to channel variations*. Lastly, we show how those POs are able to render the construction algorithm *adaptive to block length* variations.

Theorem 7. *If the above POs infer that $W_i \preccurlyeq W_j$ in a code with block-length N, then $W_i \preccurlyeq W_j$ in a code with block-length $2N$.*

Proof: A binary representation $(i_n, i_{n-1}, \cdots, i_1)$ of an index i in a code with block length $N = 2^n$, represents the same index as $(0, i_n, i_{n-1}, \cdots, i_1)$ in a code with block length $2N = 2^{n+1}$. But this lengthened representation doesn't affect the ordering given by the above POs. □

2.4 Upgraded and Degraded approximations of the bit-channels

In [7], a method was proposed, for approximating one stochastically degraded and one stochastically upgraded version of the original bit-channel. Then, the original bit-channel must lie between the degraded and the upgraded versions. In practice, the derived channels are very close and therefore the method gives very close approximations of the bit-channels.

The method runs in $O(N\mu^2 \log \mu)$, where the "fidelity" parameter μ is an even integer, and it is the quantization limit of the approximations. Basically, a greater μ gives a denser quantization, and thus closer approximations. Below we restate the theoretical background of the two algorithms. Lastly, we end this subsection by presenting the algorithms themselves.

We recall from Sections 1.3 and 1.4 that the bit-channels in polar codes can be constructed according to the binary representation of their indices, by using recursively the relations (12) and (13), whether the next bit in the representation is 1 or 0 respectively.

We will notate the equations (12) and (13) as the operations $W \circledast W$ and $W \circledcirc W$, respectively. Thus, those operations are defined as:

$$(W \circledast W)(y_1, y_2|u_1) \triangleq \frac{1}{2} \sum_{u_2 \in X} W(y_1|u_1 \oplus u_2) W(y_2|u_2) \tag{63}$$

$$(W \circledcirc W)(y_1, y_2, u_1|u_2) \triangleq \frac{1}{2} W(y_1|u_1 \oplus u_2) W(y_2|u_2). \tag{64}$$

We assume W is symmetric, and thus for every symbol $y \in Y$ in the output alphabet of W, there exists its conjugate symbol in the output alphabet, notated by $\bar{y} \in Y$, for which $W(y|1) = W(\bar{y}|0)$. In [7, Lemma 4], it is proved that we lose no information by assuming that W has no self-conjugate symbols (that is, $y = \bar{y}$). Henceforth, for simplicity we will assume that W has no self-conjugate symbols.

We also associate for each output symbol $y \in Y$ a *likelihood ratio*, defined as follows:

$$LR(y) \triangleq \frac{W(y|0)}{W(y|1)} = \frac{W(y|0)}{W(\bar{y}|0)}. \tag{65}$$

Lemma 2. *Fix a binary input channel $W : X \to Y$. Denote $W_{\circledast} = W \circledast W$ and $W_{\circledcirc} = W \circledcirc W$. Suppose that a channel Q is degraded with respect to W, and denote $Q_{\circledast} = Q \circledast Q$ and $Q_{\circledcirc} = Q \circledcirc Q$. Then*

$$Q_{\circledast} \preccurlyeq W_{\circledast} \quad \textit{and} \quad Q_{\circledcirc} \preccurlyeq W_{\circledcirc}. \tag{66}$$

Moreover, all of the above continues to hold if we replace "degraded" by "upgraded" and "$\preccurlyeq$" by "$\succcurlyeq$".

2.4.1 Degrading merge

Lemma 3. *Let $W : X \to Y$ be a BMS channel, and let y_1 and y_2 be symbols in the output alphabet Y. Define the channel $Q : X \to Z$ as follows. The output alphabet Z is given by*

$$Z = Y \setminus \{y_1, \bar{y}_1, y_2, \bar{y}_2\} \cup \{z_{1,2}, \bar{z}_{1,2}\}. \tag{67}$$

For all $x \in X$, $z \in Z$, define

$$Q(z|x) = \begin{cases} W(z|x), & \text{if } z \notin \{\bar{z}_{1,2}, z_{1,2}\}, \\ W(y_1|x) + W(y_2|x), & \text{if } z = z_{1,2}, \\ W(\bar{y}_1|x) + W(\bar{y}_2|x), & \text{if } z = \bar{z}_{1,2}. \end{cases} \tag{68}$$

Then, $Q \preccurlyeq W$.

Lemma 3 is used repetetively inbetween the recursive applications of "⊠" and "⊛". Hence, the computed bit-channel is a degraded version of the actual bit-channel, while its output alphabet size is reduced by 2 for each time Lemma 3 is used. The reduction of the output alphabet size reduces the time complexity and the memory requirements for the next recursion step. We pair Lemma 3 with the equations (20), (21) which we restate below in terms of the defined operations.

$$Z(W \boxtimes W) \leq 2Z(W) - Z(W)^2 \tag{69}$$

$$Z(W \circledast W) = Z(W)^2 \tag{70}$$

Algorithm 1 constructs a degraded version Q of a bit-channel $W_N^{(i)}$ and outputs its probability of error. This can easily be seen by noticing that Algorithm A just uses Lemma 3 and the equations (20) and (21) recursively. Because Q is degraded with respect to W, $P_e(Q)$ is an upper bound for the probability of error of $W_N^{(i)}$.

We want the degraded version of $W_N^{(i)}$ to be as close to the original. That is, its probability of error must be as low as possible, or equivalently, its capacity must be as high as possible. Thus, we must find for which pair $\{y_i, y_j\}$ the application of Lemma 3 produces a channel with the largest possible capacity.

Theorem 8. *Let $W : x \to Y$ be a BMS channel, with $Y = \{y_1, y_2, \ldots, y_L, \bar{y}_1, \bar{y}_2, \ldots, \bar{y}_L\}$. Assume that*

$$1 \leq LR(y_1) \leq LR(y_2) \leq \cdots \leq LR(y_L). \tag{71}$$

Algorithm 1 Bit-channel degrading procedure

procedure BITCHANNEL_DEGRADING(W, μ, N, i)
▷ **inputs:** An underlying BMS channel W, a bound $\mu = 2\nu$ on the output alphabet size, a code length $N = 2^n$, and an index i with binary representation $i = (b_1, b_2, \ldots, b_n)_2$.
▷ **output:** An upper bound on the probability of error of W_i.

    $\mathrm{Z} = Z(W)$
    $Q = \text{DEGRADING\_MERGE}(W, \mu)$
    for $j = 1, 2, \ldots, n$ **do**
        if $b_j = 0$ **then**
            $W \leftarrow Q \boxplus Q$
            $Z \leftarrow \text{MIN}(Z(W), 2\mathrm{Z}^2 - \mathrm{Z})$
        else
            $W \leftarrow Q \circledast Q$
            $\mathrm{Z} \leftarrow \mathrm{Z}^2$
        end if
        $Q = \text{DEGRADING\_MERGE}(W, \mu)$
    end for
    $P_e(Q) = \frac{1}{2} \sum_{y \in Y} \text{MIN}(W(y|0), W(y|1))$
    return $\text{MIN}(P_e(Q), \mathrm{Z})$

end procedure

For symbols $w_1, w_2 \in Y$, denote by $I(w_1, w_2)$ the capacity of the channel one gets by the application of Lemma 3 to w_1 and w_2. Then, for all distinct $1 \leq i \leq L$ and $1 \leq j \leq L$,

$$I(\bar{y}_i, \bar{y}_j) = I(y_i, y_j) \geq I(y_i, \bar{y}_j) = I(\bar{y}_i, y_j). \tag{72}$$

Moreover, for all $1 \leq i < j < k \leq L$, we have that either

$$I(y_i, y_j) \geq I(y_i, y_k), \tag{73}$$

or

$$I(y_j, y_k) \geq I(y_i, y_k). \tag{74}$$

Essentially, Theorem 8 says that assuming we have ordered the likelihood ratios of the conjugate output symbols as in (71), it suffices to choose the pair of the consecutive symbols, for which Lemma 3 produces a channel with the largest capacity. By doing this, we can maximize the capacity of the produced channel. This way we consider only $L - 1$ merges instead of $\binom{L}{2}$.

The contribution of a conjugate pair of output symbols $\{a, b\}$ in the overall capacity of a channel, is given by

$$C(a,b) = -(a+b)\log_2((a+b)/2) + a\log_2(a) + b\log_2(b). \tag{75}$$

We use the notation $a = W(y_i|0)$, $b = W(\bar{y}_i|0)$, $a' = W(y_{i+1}|0)$, $b' = W(\bar{y}_{i+1}|0)$ and $a^+ = a + a'$, $b^+ = b + b'$. The resulting difference in capacity when applying Lemma 3 to y_i and y_{i+1}, is given by

$$calcDeltaI(a,b,a',b') = C(a,b) + C(a',b') - C(a^+,b^+). \tag{76}$$

Algorithm 2 contains an implementation of *degrading_merge*, which uses Lemma 3 and Theorem 8. The function *degrading_merge* applies Lemma 3 as many times as is needed in order to reduce the output alphabet size to at most μ. This restricts the running time and the space requirements until the algorithm has finished running.

More precisely, this implementation uses a data structure which integrates a doubly linked list, for storing the order of the LR values, and a min-heap, for storing the order of the $deltaI$ values. The fields $dLeft$ and $dRight$ lie in the doubly linked list, and make up the pointers to the elements corresponding to the linked pairs $\{y_{i-1}, y_i\}$ and $\{y_{i+1}, y_{i+2}\}$, respectively.

The function *insertRightmost* inserts a data element as the rightmost element of the doubly linked list. The function $getMin$ lies in the min-heap and returns the data with the smallest $deltaI$. Namely, the data element which consists the symbols we are about to merge. The function $removeMin$ removes the element returned by $getMin$. The function *valueUpdated* updates the heap due to a change in $deltaI$ resulting from a merge. Whenever we remove (insert) an element, it must be removed from (inserted in) both the list and the heap, but when a merge occurs only the heap needs to be updated. This is a result of the following lemma which says that after a merge, the resulting LR order remains the same as before, and thus we don't need to update the list after a merge.

Lemma 4. *If y_i and y_{i+1}, in light of (71), are merged to z according to Lemma 3, then*

$$LR(y_i) \le LR(z) \le LR(y_{i+1}). \tag{77}$$

Having said the above, and after considering the running time of the respective heap's and list's functions, we infer that the running time of *degrading_merge* is in $O(L\log L)$. In addition, we observe that after applying the transformations in either (63) or (64), the output alphabet size L scales to either μ^2 or $2\mu^2$, respectively. We conclude that the running time of *degrading_merge* is in $O(\mu^2\log\mu^2)$.

Algorithm 2 The degrading_merge function

procedure DEGRADING_MERGE(W, μ)
▷ **inputs:** A BMS channel $W : X \to Y$, where $|Y| = 2L$, a bound $\mu = 2\nu$ on the output alphabet size.
▷ **output:** A degraded channel $Q : X \to Y'$, where $|Y'| \leq \mu$.
▷ Assume $1 \leq LR(y_1) \leq LR(y_2) \leq \cdots \leq LR(y_L)$

    if $2L \leq \mu$ **then**
        return W
    end if
    for $i = 1, 2, \ldots, L-1$ **do**
        $d =$ new data element
        $d.a \leftarrow W(y_i|0), \quad d.b \leftarrow W(\bar{y}_i|0)$
        $d.a' \leftarrow W(y_{i+1}|0), \quad d.b' \leftarrow W(\bar{y}_{i+1}|0)$
        $d.deltaI \leftarrow$ CALCDELTAI$(d.a, d.b, d.a', d.b')$
        INSERTRIGHTMOST(d)
    end for
    $l = L$
    while $l > \nu$ **do**
        $d \leftarrow$ GETMIN()
        $a^+ = d.a + d.a', \quad b^+ = d.b + d.b'$
        $dLeft = d.left$
        $dRight = d.right$
        REMOVEMIN()
        $l \leftarrow l - 1$
        if $dLeft \neq null$ **then**
            $dLeft.a' \leftarrow a^+$
            $dLeft.b' \leftarrow b^+$
            $dLeft.deltaI \leftarrow$ CALCDELTAI$(dLeft.a, dLeft.b, a^+, b^+)$
            VALUEUPDATED$(dLeft)$
        end if
        if $dRight \neq null$ **then**
            $dRight.a \leftarrow a^+$
            $dRight.b \leftarrow b^+$
            $dRight.deltaI \leftarrow$ CALCDELTAI$(a^+, b^+, dRight.a', dRight.b')$
            VALUEUPDATED$(dRight)$
        end if
    end while
    Construct Q according to the probabilities in the data structure and return.

end procedure

Algorithm 3 Bit-channel upgrading procedure

procedure BITCHANNEL_DEGRADING(W, μ, N, i)
▷ **inputs:** An underlying BMS channel W, a bound $\mu = 2\nu$ on the output alphabet size, a code length $N = 2^n$, and an index i with binary representation $i = (b_1, b_2, \ldots, b_n)_2$.
▷ **output:** A lower bound on the probability of error of W_i.

    $Q =$ UPGRADING_MERGE(W, μ)
    for $j = 1, 2, \ldots, n$ **do**
        if $b_j = 0$ **then**
            $W \leftarrow Q \boxplus Q$
        else
            $W \leftarrow Q \circledast Q$
        end if
        $Q =$ UPGRADING_MERGE(W, μ)
    end for
    $P_e(Q) = \frac{1}{2} \sum_{y \in Y}$ MIN$(W(y|0), W(y|1))$
    return $P_e(Q)$

end procedure

2.4.2 Upgrading merge

Algorithm 3 contains the procedure for the construction of an upgraded version Q of a bit-channel $W_N^{(i)}$, which is similar to the degrading procedure. Next, we will show how its "core" works. Namely, how we can merge output symbols and get an upgraded channel instead. For the *upgrading_merge* function we employ the following two lemmas.

Lemma 5. *Let $W : X \to Y$ be a BMS channel, and let y_2 and y_1 be symbols in the output alphabet Y. Denote $\lambda_2 = LR(y_2)$ and $\lambda_1 = LR(y_2)$. Assume that*

$$1 \leq \lambda_1 \leq \lambda_2. \tag{78}$$

Next, let $a_1 = W(y_1|0)$ and $b_1 = W(\bar{y}_1|0)$. Define α_2 and β_2 as follows. If $\lambda_2 < \infty$

$$\alpha_2 = \lambda_2 \frac{a_1 + b_1}{\lambda_2 + 1}, \quad \beta_2 = \frac{a_1 + b_1}{\lambda_2 + 1}. \tag{79}$$

Otherwise, we have $\lambda_2 = \infty$, and so define

$$\alpha_2 = a_1 + b_1, \quad \beta_2 = 0. \tag{80}$$

For real numbers α, β, and $x \in X$, define

$$t(\alpha, \beta) = \begin{cases} \alpha, & \text{if } x = 0, \\ \beta, & \text{if } x = 1. \end{cases} \tag{81}$$

Define the channel $Q' : X \to Z'$ as follows. The output alphabet Z' is given by

$$Z' = Y \backslash \{y_2, \bar{y}_2, y_1, \bar{y}_1\} \cup \{z_2, \bar{z}_2\}. \tag{82}$$

For all $x \in X$ and $z \in Z'$,

$$Q'(z|x) = \begin{cases} W(z|x), & \text{if } z \notin \{z_2, \bar{z}_2\}, \\ W(y_2|x) + t(\alpha_2, \beta_2|x), & \text{if } z = z_2, \\ W(\bar{y}_2|x) + t(\beta_2, \alpha_2|x), & \text{if } z = \bar{z}_2. \end{cases} \tag{83}$$

Then, $Q' \succcurlyeq W$.

Lemma 6. *Let $W : X \to Y$ be a BMS channel, and let y_1, y_2, y_3 be symbols in the output alphabet Y. Denote $\lambda_1 = LR(y_1)$, $\lambda_2 = LR(y_2)$, $\lambda_3 = LR(y_3)$. Assume that*

$$1 \leq \lambda_1 < \lambda_2 < \lambda_3. \tag{84}$$

Next, let $a_2 = W(y_2|0)$ and $b_2 = W(\bar{y}_2|0)$. Define $\alpha_1, \beta_1, \alpha_3, \beta_3$ as follows. If $\lambda_3 < \infty$

$$\alpha_1 = \lambda_1 \frac{\lambda_3 b_2 - a_2}{\lambda_3 - \lambda_1}, \quad \beta_1 = \frac{\lambda_3 b_2 - \alpha_2}{\lambda_3 - \lambda_1}, \tag{85}$$

$$\alpha_3 = \lambda_3 \frac{a_2 - \lambda_1 b_2}{\lambda_3 - \lambda_1}, \quad \beta_3 = \frac{a_2 - \lambda_1 b_2}{\lambda_3 - \lambda_1}. \tag{86}$$

Otherwise, we have $\lambda_3 = \infty$, and so define

$$\alpha_1 = \lambda_1 b_2, \quad \beta_1 = b_2, \tag{87}$$

$$\alpha_3 = a_2 - \lambda_1 b_2, \quad \beta_3 = 0. \tag{88}$$

Leta $t(\alpha, \beta|x)$ be defined as in Lemma 5, and define the BMS channel $Q' : X \to Z'$ as follows. The output alphabet Z' is given by

$$Z' = Y \backslash \{y_1, \bar{y}_1, y_2, \bar{y}_2, y_3, \bar{y}_3\} \cup \{z_1, \bar{z}_1, z_3, \bar{z}_3\}. \tag{89}$$

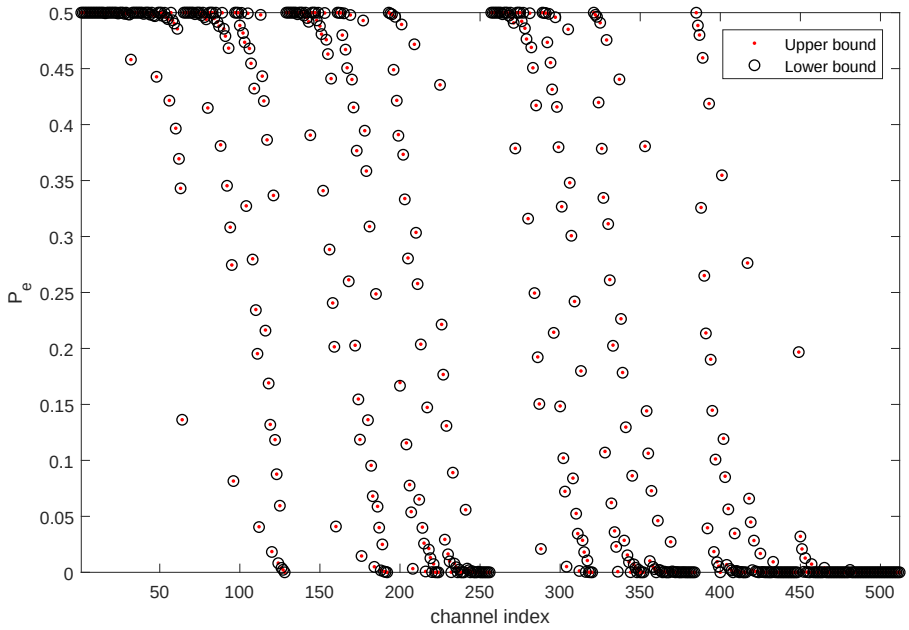

Figure 15: Upper and lower bounds of the probability of error for the BSC (N = 512, I(W) = 0.5). Chosen value of μ is 32.

For all $x \in X$ and $z \in Z'$, define

$$Q'(z|x) = \begin{cases} W(z|x), & \text{if } z \notin \{z_1, \bar{z}_1, z_3, \bar{z}_3\}, \\ W(y_1|x) + t(\alpha_1, \beta_1|x), & \text{if } z = z_1, \\ W(\bar{y}_1|x) + t(\beta_1, \alpha_1|x), & \text{if } z = \bar{z}_1, \\ W(y_3|x) + t(\alpha_3, \beta_3|x), & \text{if } z = z_3, \\ W(\bar{y}_3|x) + t(\beta_3, \alpha_3|x), & \text{if } z = \bar{z}_3. \end{cases} \tag{90}$$

Then, $Q' \succcurlyeq W$.

Regarding how Lemmas 5 and 6 perform compared to each other, in [7, Lemma 12] it is proved that Lemma 6 produces a channel that is closer to the original bit-channel than Lemma 5 does. The reason Lemma 5 is used at all, is because when λ_1 and λ_3 are too close to each other, the subtraction operations in Lemma 6 will cause numerical instabilities when performed from a logical floating-point machine. For this reason, Lemma 5 is used instead of Lemma 6 when this case occurs.

The merge-upgrading procedure follows the same structure as the degrading-merge. The only twist here is that we must first search for which indices $1 \leq i \leq L-1$ the ratio

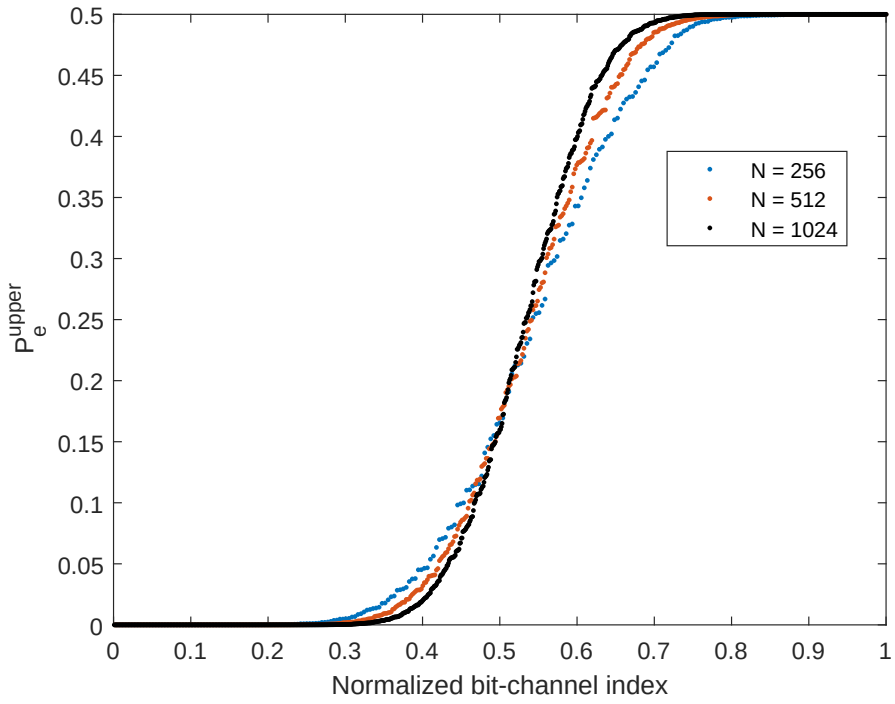

Figure 16: Polarization effect of the BSC (N = 512, I(W) = 0.5).

$LR(y_{i+1})/LR(y_i)$ is less than $1+\epsilon$ for small ϵ. Namely, the symbols for which their LR is too close. For these symbols, we must use Lemma 5, until no such index exists, or until the output alphabet size is at most μ. Then we continue by applying Lemma 6 for the rest merging procedure.

The order we choose the output symbols for applying the merging operations is, again, that which minimizes the capacity deviation from the initial channel. For Lemma 5, this is the same as that which is used in the degrading-merge procedure (76). For Lemma 6 we similarly define the resulting difference in capacity as follows: Let a_2, b_2, α_1, β_1, α_3, β_3 be defined according to Lemma 6. Also, let $a_1 = W(y_1)$, $b_1 = W(\bar{y}_1)$, $a_3 = W(y_3)$, $b_3 = W(\bar{y}_3)$ and for shorthand we notate $a_1^+ = a_1 + \alpha_1$, $b_1^+ = b_1 + \beta_1$, $a_3^+ = a_3 + \alpha_3$, $b_3^+ = b_3 + \beta_3$. Then, the resulting difference in capacity is given by

$$\begin{aligned} calcDeltaI_6(a_1, b_1, a_2, b_2, a_3, b_3, a_1^+, b_1^+, a_3^+, b_3^+) =& (a_1, b_1) + C(a_2, b_2) + C(a_3, b_3) \\ & - C(a_1^+, b_1^+) - C(a_3^+, b_3^+), \end{aligned} \tag{91}$$

where the subscript of the function indicates that it refers to Lemma 6.

Figure 15 illustrates the approximated upper and lower bounds for the BSC, using a fidelity parameter $\mu = 32$. Figure 16 illustrates the polarization effect of the same

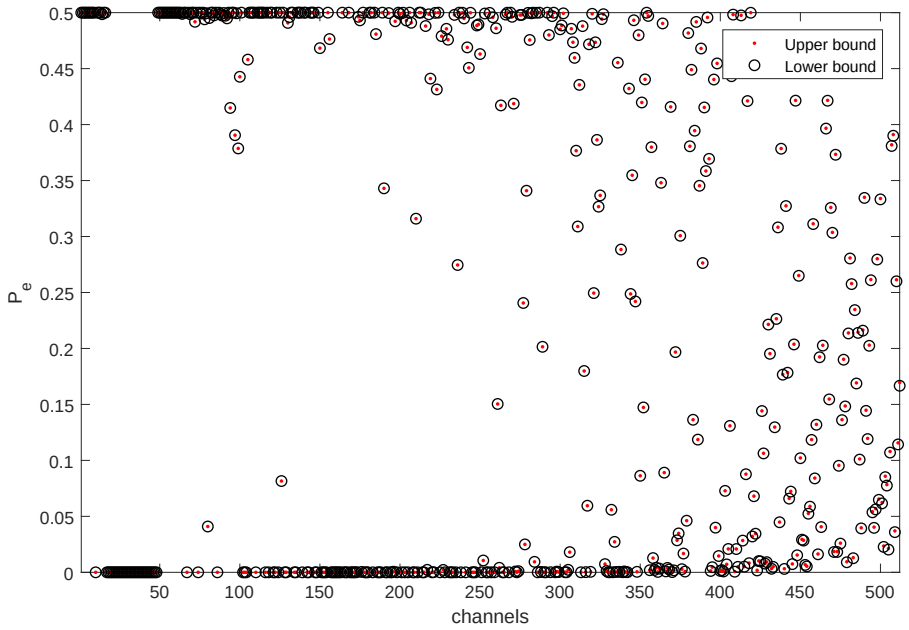

Figure 17: Bit-channels of the BSC (N = 512, I(W) = 0.5), sorted according to the differences $\{P_e^{upper}(W_N^{(i)}) - P_e^{lower}(W_N^{(i)})\}$ (right is higher), for $\mu = 32$. The measurements ranged in $[0, 2.86\text{e}{-03})$, with mean value 8.5e−05 and variance 5.35e−08. By evaluating only on the best half channels, the mean value is 7.45e−05 and the variance is 6.19e−08.

channel by sorting the upper bound of probability of error. We can easily notice that as the block length N increases, the fraction of the bit-channels that have extremal (bounds of) probability of error increases, too.

2.4.3 Performance of the Approximations

In our proposed algorithm, we will assume that the inequality $P_e^{upper}(W_i) \leq P_e^{upper}(W_j)$ infers the inequality $P_e(W_i) \leq P_e(W_j)$. Below, we study the safety of this assumption.

Firstly, we study the following observation: if the distance between the upper and lower bounds of the probability of error of a bit-channel is very small, then the bounds are very close to the true probability. If the same happens for two different bit-channels we can assume that the bounds are able to order their actual probability of error correctly. In other words, if the differences $\left(P_e^{upper}(W_i) - P_e^{lower}(W_i)\right)$ and $\left(P_e^{upper}(W_j) - P_e^{lower}(W_j)\right)$ are very small, then $P_e^{upper}(W_i) \leq P_e^{upper}(W_j)$ infers that $P_e(W_i) \leq P_e(W_j)$. The same holds if we switch "$\leq$" with "$\geq$". In Figure 17 we sort the bit-channels according to this measure. We observe that the distance is higher for the channels that are not yet

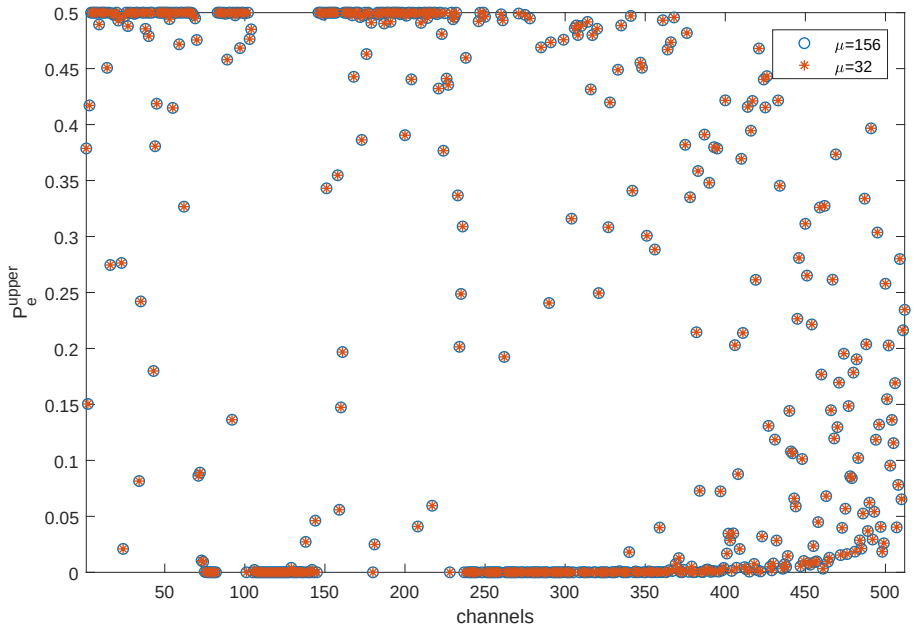

Figure 18: Bit-channels of the BSC (N = 512, I(W) = 0.5), sorted according to the differences $\{P_e^{upper,average\ \mu}(W_N^{(i)}) - P_e^{upper,high\ \mu}(W_N^{(i)})\}$ (right is higher). The measurements ranged in $[0, 2.59\text{e}{-}04)$, with mean value 1.48e−05 and variance 1.64e−09. By evaluating only on the best half channels, the mean value is 9.61e−06 and the variance is 1.16e−09.

polarized. Because polar codes are not likely to use these channels, we infer that the above assumption is safe.

We insist in our assumption by studying an additional measure of the approximation performance: if for two different bit-channels their upper bound of probability of error for some average value of μ, is close to that for a relatively high value of μ, then we infer that the bounds for the average value of μ are already close to the true values. Hence, we can assume that they are able to order them correctly. In Figure 18 we sort the bit-channels according to this measure. We make the same observation as before: the difference is higher for the intermediate channels.

A general implication of Figures 17 and 18 is that Algorithms 1 and 3 work better mostly for those bit-channels that are nearly polarized. We are now ready to present the proposed construction algorithm.

2.5 Fast construction of Polar Codes

We begin by determining all the possible $\binom{N}{2}$ pair-wise orderings of the bit-channels using the available Partial Orders. We keep these orderings in a matrix, called *InformationTable* (IT) as follows: If $W_i \preccurlyeq W_j$, then $IT(i,j) = -1$. If $W_j \preccurlyeq W_i$, then $IT(i,j) = 1$. For any other case, $IT(i,j) = 0$. We make the following observations regarding the structure of IT.

With a first look we see that IT is an $N \times N$ matrix. By observing that $IT(i,i) = 0$ for any i, IT can be reduced to a $(N-1) \times N$ matrix. Also, by observing that $IT(i,j) = r \Leftrightarrow IT(j,i) = -r,\ r \in \{-1,0,1\}$, IT can be further reduced to a vector of length $(N-1)N/2$. Lastly, keeping this structure, IT's elements take only binary values $\{0,1\}$. Thus, in order to represent IT's values, it is only needed 1 bit per element. We infer that saving IT requires only $(N^2-N)/2$ bits. However, while referring to IT we will keep the original $N \times N$ structure because it is easier to depict.

Also, in use cases where the block length may vary dynamically, we still only need one static IT, with size equal to the largest block-length that our module uses. This holds true because of Theorem 7. For example, if our maximum block-length is N_{max} and we wish to construct a polar code of length $N_k \leq N_{max}$, we will just use the upper-left $N_k \times N_k$ part of $IT_{N_{max} \times N_{max}}$.

Because of the above, and because the values of IT are independent of the physical channel, IT is needed to be calculated only once, and use it repetitively whenever we need to construct a polar code. Thus, we will assume that the construction of IT doesn't contribute to the running time of the algorithm.

Having said the above, we assume IT is already calculated. The construction procedure begins by using IT for calculating the subsets I, F and U. We then use Algorithm 1 in Section 2.4 to calculate upper bounds of probability of error for the bit-channels in U. Then, we sort the bit-channels in U according to their upper probability of error and choose the best of them to complete A and the worst to complete A^c. Algorithm 4 is indicative of this procedure. Also, it is easy to integrate the adaptive properties of the partial orders that we mentioned in Chapter 2.3.

Example: Consider the BSC W, with I(W) = 0.5. Suppose we wish to construct a polar code for W, with block length N = 16 and rate R = 0.44. Then, the information set must have size equal to K = $\lfloor N \cdot R \rfloor$ = 7. The given partial orders yield the following Information Table (IT).

Algorithm 4 Proposed construction method of Polar Codes

procedure CONSTRUCT(W, μ, N, R)
▷ **inputs:** An underlying BMS channel W, a bound $\mu = 2\nu$ on the output alphabet size for the degrading/upgrading procedures, a code length $N = 2^n$, and a transmission rate R.
▷ **outputs:** The information set A and the frozen vector A^c.

$K \leftarrow \lfloor N \cdot R \rfloor$
$I = \{i : |IT(i,:) == 1| \geq N - K\}$
$F = \{j : |IT(j,:) == -1| \geq K\}$
$U = \{k : k \notin I \cup F\}$

Run Algorithm 1 on the channels in U and return the upper bounds of their probability of error in the vector PeU.
Sort PeU and save in C the permutation vector.

▷ Now C contains the indices of the channels in U, sorted by their upper bounds of probability of error.

$A = I \cup C(1 : K - |I|)$
return A and A^c

end procedure

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	0	−1	−1	−1	−1	−1	−1	−1	−1	−1	−1	−1	−1	−1	−1	−1
1	1	0	−1	−1	−1	−1	−1	−1	−1	−1	−1	−1	−1	−1	−1	−1
2	1	1	0	−1	−1	−1	−1	−1	−1	−1	−1	−1	−1	−1	−1	−1
3	1	1	1	0	0	−1	−1	−1	0	−1	−1	−1	0	−1	−1	−1
4	1	1	1	0	0	−1	−1	−1	−1	−1	−1	−1	−1	−1	−1	−1
5	1	1	1	1	1	0	−1	−1	0	−1	−1	−1	−1	−1	−1	−1
6	1	1	1	1	1	1	0	−1	0	0	−1	−1	−1	−1	−1	−1
7	1	1	1	1	1	1	1	0	0	0	0	−1	0	−1	−1	−1
8	1	1	1	0	1	0	0	0	0	−1	−1	−1	−1	−1	−1	−1
9	1	1	1	1	1	1	0	0	1	0	−1	−1	−1	−1	−1	−1
10	1	1	1	1	1	1	1	0	1	1	0	−1	−1	−1	−1	−1
11	1	1	1	1	1	1	1	1	1	1	1	0	0	−1	−1	−1
12	1	1	1	0	1	1	1	0	1	1	1	0	0	−1	−1	−1
13	1	1	1	1	1	1	1	1	1	1	1	1	1	0	−1	−1
14	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	−1
15	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0

where the indices of the rows and columns correspond to indices of bit-channels, and the cells take values as we described above. From this matrix, we construct the subsets I, F and U as

$$\begin{aligned} I =& \{i : |IT(i,:) == 1| \geq 9\} = \{10, 11, 12, 13, 14, 15\}, \\ F =& \{j : |IT(j,:) == -1| \geq 7\} = \{0, 1, 2, 3, 4, 5, 6, 8\}, \\ U =& \{k : k \notin I \cup F\} = \{7, 9\}. \end{aligned}$$

The cardinality of I is 6. Thus, we need 1 more bit-channel to complete A. The output of Algorithm 1 ($\mu = 32$), for the bit-channels with indices in U, is $P_e^{upper}(W_{16}^{(7)}) \simeq 0.03099$ and $P_e^{upper}(W_{16}^{(9)}) \simeq 0.20726$. Thus, the bit-channel with index 7 is more reliable than the bit-channel with index 9, and the final information set of the polar code is

$$A = I \cup \{7\} = \{7, 10, 11, 12, 13, 14, 15\}.$$

Lastly, we note that we could combine the pair-wise orderings from the partial orders with the newly mined from using Algorithm 1: after running Algorithm 1 we could fill some of the gaps (zeros) in IT using the sorted vector PeU. Then we could further use the transitivity property (57) in order to fill more gaps: for any i, j, k, if $IT(i, j) = 1$ and $IT(j, k) = 1$, then $IT(i, k) = 1$. Similarly, if $IT(i, j) = -1$ and $IT(j, k) = -1$, then $IT(i, k) = -1$. We then could repeat the procedure and get updated versions of I, F and U. Our experiments showed that although we can indeed fill some part of the gaps this way, the produced information sets using this method are the same as before regardless of the chosen fidelity parameter μ, and thus the increased time complexity is needless.

2.6 Results

In this section we present the reliability and efficiency performance of the proposed algorithm. Also, we will compare how the proposed algorithm performs with respect to the conventional method that chooses the best channels by sorting the upper bounds of probability of error without employing the partial orders of the bit-channels.

One question that arises is how the fidelity parameter μ affects the resulting information sets. in Section 2.1.1, we observed that the fixed-rate problem should be faster to solve because we only need to order the reliability of the channels and not to precisely calculate them. In practice this comes true. We take for example a BSC with I(W) = 0.5 and

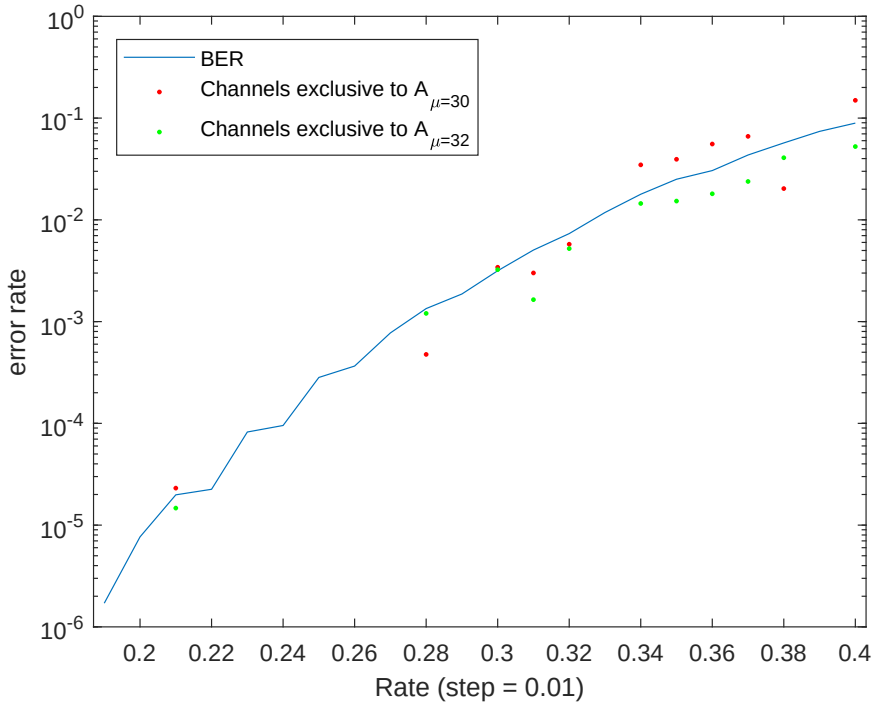

Figure 19: Error rate performance for the BSC (N = 512, I(W) = 0.5).

block-length N = 512. In this case, the information sets produced by sorting the upper bounds of probability of error are exactly the same for any value of μ when $\mu \geq 32$. This means that we lose nothing by restricting μ to at most 32 for this channel and block-length setting.

As for the difference between the produced information sets of the proposed algorithm and the conventional method described above, it turns out that, for reasonably high values of μ (e.g. in our example for $\mu \geq 4$), the resulting information sets are exactly the same when the two methods run for the same μ.

We illustrate the proposed algorithm's performance in Figures 19 and 20. It turns out that the proposed method can use higher values of μ and still maintain a faster running time than the conventional method, regardless of the increased value of μ. In our example, the resulting information sets differ to at most 1 bit-channel. However, because in practice channel polarization is constrained by the block length, the assumption we made in Section 1.6 that, successive cancellation decoding can reliably assume that, when estimating the input u_i, our estimations of all the previous inputs u_1^{i-1} are errorless, is not absolutely true. Hence, in polar codes there exist error propagation when the estimation of an input is wrong and this estimation is used when decoding other inputs. For this reason,

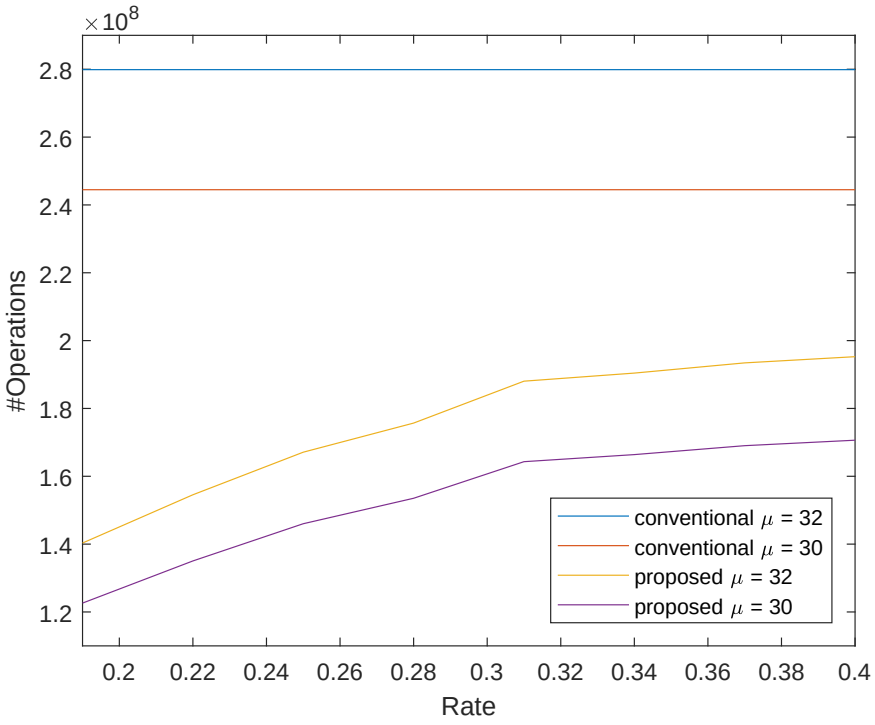

Figure 20: Number of numerical operations for the BSC (N = 512, I(W) = 0.5).

when we use a more reliable bit-channel, with input u_i, we also render more reliable the bit-channels that have u_i in their output vector. Having said that, it is important to note that, albeit minimal in population, the different channels that our proposed algorithm gives are most of the times better, as a result from using a greater fidelity parameter.

In summary, for the same error rate performance, or equivalently for the same value of the fidelity parameter, the proposed algorithm is significantly faster. On the other hand, if we wish to constrain the running time, we can use higher values of the fidelity parameter and get more accurate approximations, getting slightly more reliable information sets.

Appendix

Proof of Proposition 1.

To prove (4), we use Gallager's $E_0(p, Q)$ [2, p.138], which can be seen as a measure of *information*:

$$E_0(p, Q) \doteq -\log \sum_{y \in Y} \left[\sum_{x \in X} Q(x) W(y|x)^{1/(1+p)} \right]^{1+p}, \quad 0 \leq p \leq 1. \tag{92}$$

When Q is the uniform input distribution and $p = 1$, we have

$$E_0(1, Q) = -\log \sum_{y \in Y} \left[\frac{1}{2} \left(\sqrt{W(y|0)} + \sqrt{W(y|1)} \right) \right]^2 = \log \frac{2}{1 + Z(W)}. \tag{93}$$

Also, in [2, Theorem 5.6.3] it is shown that $I(W) \geq \frac{\partial E_0(p,Q)}{\partial p}$. We use this to show:

$$\frac{\partial I(W)p}{\partial p} \geq \frac{\partial E_0(p, Q)}{\partial p}. \tag{94}$$

Also, notice that $E_0(p, Q) = 0$ and $I(W)p = 0$ when $p = 0$. Then, we have

$$I(W)p \geq E_0(p, Q), \ \ \forall\ p \geq 0. \tag{95}$$

If we set $p = 1$ in (95), we get inequality (4). Inequality (5) is proved in [1, Appendix]. □

Proof of the tranformation relations (15) and (16).

To prove (15), we write

$$\begin{aligned} W_{2N}^{(2i-1)}(y_1^{2N}, u_1^{2i-2}|u_{2i-1}) &= \sum_{u_{2i}^{2N} \in X^{2N-2i}} \frac{1}{2^{2N-1}} W_{2N}(y_1^{2N}|u_1^{2N}) \\ &= \sum_{u_{2i,o}^{2N}, u_{2i,e}^{2N}} \frac{1}{2^{2N-1}} W_N(y_1^N|u_{1,o}^{2N} \oplus u_{1,e}^{2N}) W_N(y_{N+1}^{2N}|u_{1,e}^{2N}) \\ &= \sum_{u_{2i}} \frac{1}{2} \sum_{u_{2i+1,e}^{2N}} \frac{1}{2^{N-1}} W_N(y_{N+1}^{2N}|u_{1,e}^{2N}) \\ &\quad \cdot \sum_{u_{2i+1,o}^{2N}} \frac{1}{2^{N-1}} W_N(y_1^N|u_{1,o}^{2N} \oplus u_{1,e}^{2N}) \end{aligned} \tag{96}$$

By definition (9), the sum over $u_{2i+1,o}^{2N}$ for any fixed $u_{1,e}^{2N}$ equals $W_N^{(i)}(y_1^N, u_{1,o}^{2i-2} \oplus u_{1,e}^{2i-2} | u_{2i-1} \oplus u_{2i})$, because as $u_{2i+1,o}^{2N}$ ranges over X^{N-i}, $u_{2i+1,o}^{2N} \oplus u_{2i+1,e}^{2N}$ ranges also over X^{N-i}. We now factor this term out of the middle sum in the above equation and use (9) again to obtain (15). To prove (16), we write

$$\begin{aligned} W_{2N}^{(2i)}(y_1^{2N}, u_1^{2i-1} | u_{2i}) = & \sum_{u_{2i+1,e}^{2N}} \frac{1}{2^{2N-1}} W_{2N}(y_1^{2N} | u_1^{2N}) \\ = & \frac{1}{2} \sum_{u_{2i+1,e}^{2N}} \frac{1}{2^{N-1}} W_N(y_{N+1}^{2N} | u_{1,e}^{2N}) \\ & \cdot \sum_{u_{2i+1,o}^{2N}} \frac{1}{2^{N-1}} W_N(y_1^N | u_{1,o}^{2N} \oplus u_{1,e}^{2N}). \end{aligned} \tag{97}$$

By carrying out the inner and outer sums in the same manner as in the proof of (15), we obtain (16). □

Proof of Proposition 2.

Consider the channels $W : X \to Y$, $W' : X \to \tilde{Y}$ and $W'' : X \to \tilde{Y} \times X$, where $\tilde{Y} = (Y_1, Y_2)$. Define the uniformly distributed pair (U_1, U_2) over X^2 and $(X_1, X_2) = (U_1 \oplus U_2, U_2)$. Also, we define $P_{Y_1,Y_2|X_1,X_2}(y_1, y_2 | x_1, x_2) = W(y_1|x_1)W(y_2|x_2)$. The latter definition fits in our framework because (i) given X_i, Y_i is independent of any other input and (ii) given both X_1 and X_2, Y_1 and Y_2 are independent, because we use two *independent* copies of W to transmit X_1 and X_2. We now have

$$I(W') = I(U_1; Y_1 Y_2), \tag{98}$$

$$I(W'') = I(U_2; Y_1 Y_2 U_1) = I(U_2; Y_1 Y_2 | U_1), \tag{99}$$

where in the last equation we used the independence of U_1 and U_2. By the chain rule and the fact that there is a one-to-one relation between (X_1, X_2) and (U_1, U_2) we have

$$I(W') + I(W'') = I(U_1 U_2; Y_1 Y_2) = I(X_1 X_2; Y_1 Y_2). \tag{100}$$

Next, we prove a useful lemma.

Lemma 7. *$(X_1, X_2) = (U_1 \oplus U_2, U_2)$ is a pair of independent random variables.*

Proof: Notice that

$$P(u_1 \oplus u_2) = \frac{1}{2} \ \forall \ (u_1, u_2), \tag{101}$$

$$P(u_2) = \frac{1}{2} \ \forall \ u_2, \tag{102}$$

$$P(u_1 \oplus u_2, u_2) = \frac{1}{4} \ \forall \ (u_1, u_2). \tag{103}$$

Then,

$$P(u_1 \oplus u_2, u_2) = P(u_1 \oplus u_2)P(u_2) \ \forall \ (u_1, u_2). \tag{104}$$

□

We now have

$$\begin{aligned} I(X_1X_2; Y_1Y_2) &= I(X_1; Y_1Y_2) + I(X_2; Y_1Y_2|X_1) \\ &= I(X_1; Y_1Y_2) + H(X_2|X_1) - H(X_2|Y_1Y_2X_1) \\ &= I(X_1; Y_1Y_2) + H(X_2) - H(X_2|Y_1Y_2) \\ &= I(X_1; Y_1Y_2) + I(X_2; Y_1Y_2) \end{aligned} \tag{105}$$

where in the first equation we used the chain rule and in the third equation we used Lemma 7. Also,

$$\begin{aligned} I(X_1; Y_1Y_2) &= H(X_1) - H(X_1|Y_1Y_2) \\ &= H(X_1) - H(X_1|Y_1) \\ &= I(X_1; Y_1) \end{aligned} \tag{106}$$

where in the second equation we used the independence of X_1 and Y_2. Similarly,

$$I(X_2; Y_1Y_2) = I(X_2; Y_2). \tag{107}$$

The proof of (17) is now completed, since

$$(105) \overset{\substack{(106)\\(107)}}{\Longrightarrow} I(X_1X_2; Y_1Y_2) = I(X_1; Y_1) + I(X_2; Y_2) = I(W) + I(W) = 2I(W). \tag{108}$$

To prove (18), we begin by noting that

$$\begin{aligned} I(W'') =& I(U_2; Y_1Y_2U_1) = I(U_2; Y_2) + I(U_2; Y_1U_1|Y_2) \\ =& I(W) + I(U_2; Y_1U_1|Y_2) \geq I(W). \end{aligned} \tag{109}$$

(109) and (17) give (18). The above proof shows that equality holds in (18) iff $I(U_2; Y_1U_1|Y_2) = 0$, which is equivalent to having

$$P_{U_1,U_2,Y_1|Y_2}(u_1, u_2, y_1|y_2) = P_{U_1,Y_1|Y_2}(u_1, y_1|y_2)P_{U_2|Y_2}(u_2|y_2) \tag{110}$$

for all (u_1, u_2, y_1, y_2) such that $P_{Y_2}(y_2) > 0$, or equivalently, by multiplying both sides with $P_{Y_2}(y_2)^2/(P_{U_1}(u_1)P_{U_2}(u_2))$,

$$P_{Y_1,Y_2|U_1,U_2}(y_1, y_2|u_1, u_2)P_{Y_2}(y_2) = P_{Y_1,Y_2|U_1}(y_1, y_2|u_1)P_{Y_2|U_2}(y_2|u_2) \tag{111}$$

for all (u_1, u_2, y_1, y_2). Since $P_{Y_1,Y_2|U_1,U_2}(y_1, y_2|u_1, u_2) = W(y_1|u_1 \oplus u_2)W(y_2|u_2)$, (111) can be written as

$$W(y_2|u_2)[W(y_1|u_1 \oplus u_2)P_{Y_2}(y_2) - P_{Y_1,Y_2}(y_1, y_2|u_1)] = 0. \tag{112}$$

Substituting

$$P_{Y_2}(y_2) = \frac{1}{2}W(y_2|u_2) + \frac{1}{2}W(y_2|u_2 \oplus 1) \tag{113}$$

and

$$P_{Y_1,Y_2|U_1}(y_1, y_2|u_1) = \frac{1}{2}W(y_1|u_1 \oplus u_2)W(y_2|u_2) + \frac{1}{2}W(y_1|u_1 \oplus u_2 \oplus 1)W(y_2|u_2 \oplus 1) \tag{114}$$

into (112) and simplifying, we obtain

$$W(y_2|u_2)W(y_2|u_2 \oplus 1)[W(y_1|u_1 \oplus u_2) - W(y_1|u_1 \oplus u_2 \oplus 1)] = 0, \tag{115}$$

which for all possible values of (u_1, u_2) is equivalent to

$$W(y_2|0)W(y_2|1)[W(y_1|0) - W(y_1|1)] = 0. \tag{116}$$

Thus, either there exists no y_2 such that $W(y_2|0)W(y_2|1) > 0$, in which case $I(W) = 1$, or for all y_1 we have $W(y_1|0) = W(y_1|1)$, which implies $I(W) = 0$. This concludes the proof of Proposition 2. □

Proof of Proposition 3.

To prove (20), we write

$$\begin{aligned} Z(W'') &= \sum_{y_1^2,u_1} \sqrt{W''(y_1^2,u_1|0)} \cdot \sqrt{W''(y_1^2,u_1|1)} \\ &= \sum_{y_1^2,u_1} \frac{1}{2}\sqrt{W(y_1|u_1)W(y_2|0)} \cdot \sqrt{W(y_1|u_1\oplus 1)W(y_2|1)} \\ &= \sum_{y_2} \sqrt{W(y_2|0)W(y_2|1)} \cdot \sum_{u_1} \frac{1}{2} \sum_{y_1} \sqrt{W(y_1|u_1)W(y_1|u_1\oplus 1)} \\ &= Z(W)^2. \end{aligned} \tag{117}$$

To prove (21), we use the notation $\alpha(y_1) = W(y_1|0)$, $\delta(y_1) = W(y_1|1)$, $\beta(y_2) = W(y_2|0)$, $\gamma(y_2) = W(y_2|1)$.

$$\begin{aligned} Z(W') &= \sum_{y_1^2} \sqrt{W'(y_1^2|0)W'(y_1^2|1)} \\ &= \sum_{y_1^2} \frac{1}{2}\sqrt{\alpha(y_1)\beta(y_2)+\delta(y_1)\gamma(y_2)} \cdot \sqrt{\alpha(y_1)\gamma(y_2)+\delta(y_1)\beta(y_2)} \end{aligned} \tag{118}$$

Also, we use the following identity to get:

$$\begin{aligned} &[\sqrt{(\alpha\beta+\delta\gamma)(\alpha\gamma+\delta\beta)}]^2 + 2\sqrt{\alpha\beta\delta\gamma}(\sqrt{\alpha}-\sqrt{\delta})^2(\sqrt{\beta}-\sqrt{\gamma})^2 \\ &\qquad = [(\sqrt{\alpha\beta}+\sqrt{\delta\gamma})(\sqrt{\alpha\gamma}+\sqrt{\delta\beta}) - 2\sqrt{\alpha\beta\delta\gamma}]^2 \\ &\Rightarrow [\sqrt{(\alpha\beta+\delta\gamma)(\alpha\gamma+\delta\beta)}]^2 \le [(\sqrt{\alpha\beta}+\sqrt{\delta\gamma})(\sqrt{\alpha\gamma}+\sqrt{\delta\beta}) - 2\sqrt{\alpha\beta\delta\gamma}]^2 \\ &\Rightarrow [\sqrt{(\alpha\beta+\delta\gamma)(\alpha\gamma+\delta\beta)}] \le [(\sqrt{\alpha\beta}+\sqrt{\delta\gamma})(\sqrt{\alpha\gamma}+\sqrt{\delta\beta}) - 2\sqrt{\alpha\beta\delta\gamma}]. \end{aligned} \tag{119}$$

Then, from (118) we get

$$\begin{aligned} Z(W') \le &\sum_{y_1^2} \frac{1}{2}[\sqrt{\alpha(y_1)\beta(y_2)}+\sqrt{\delta(y_1)\gamma(y_2)}] \cdot [\sqrt{\alpha(y_1)\gamma(y_2)}+\sqrt{\delta(y_1)\beta(y_2)}] \\ &- \sum_{y_1^2} \sqrt{\alpha(y_1)\beta(y_2)\delta(y_1)\gamma(y_2)}]. \end{aligned} \tag{120}$$

Now, each term obtained after expanding $(\sqrt{\alpha(y_1)\beta(y_2)}+\sqrt{\delta(y_1)\gamma(y_2)})(\sqrt{\alpha(y_1)\gamma(y_2)}+\sqrt{\delta(y_1)\beta(y_2)})$ gives $Z(W)$ when summed over y_1^2. Also, $\sum_{y_1^2}\sqrt{\alpha(y_1)\beta(y_2)\delta(y_1)\gamma(y_2)} = Z(W')$. Hence the inequality is proved.

To prove the equality condition in (21), we notice that the inequality was formed in (119) when we omitted the term $\left(2\sqrt{\alpha\beta\delta\gamma}(\sqrt{\alpha}-\sqrt{\delta})^2(\sqrt{\beta}-\sqrt{\gamma})^2\right)$. Hence, we have equality in

(21) iff $\left(2\sqrt{\alpha(y_1)\beta(y_2)\delta(y_1)\gamma(y_2)}(\sqrt{\alpha(y_1)}-\sqrt{\delta(y_1)})^2(\sqrt{\beta(y_2)}-\sqrt{\gamma(y_2)})^2\right)=0$, i.e. iff for any choice of y_1^2, $\alpha(y_1)\beta(y_2)\delta(y_1)\gamma(y_2)=0$ or $\alpha(y_1)=\delta(y_1)$ or $\beta(y_2)=\gamma(y_2)$. Now we will explain why this condition is satisfied iff W is a BEC.

Suppose W is a BEC. Then, when either y_1 or y_2 is equal to 0 or 1, at least one term in $\alpha(y_1)\beta(y_2)\delta(y_1)\gamma(y_2)$ is equal to zero. If $y_1=\varepsilon$ then $\alpha(y_1)=\delta(y_1)$ and if $y_2=\varepsilon$ then $\beta(y_2)=\gamma(y_2)$. We conclude that if W is BEC, equality in (21) holds. Conversely, take the possible case that $y_1=y_2$. If in this case equality is satisfied *only* when W is a BEC, then W must be a BEC. In this case, for the equality to hold we must have, for any choice of y_1, either $\alpha(y_1)\delta(y_1)=0$ or $\alpha(y_1)=\delta(y_1)$. Comparing with the transition probabilities of a BEC (Figure 1), we notice that this is equivalent to saying that W is a BEC. We conclude that equality in (21) holds iff W is a BEC.

To prove (22), we use the following result, which is proved in [1, Lemma 4]:
Given any collection of B-DMCs $W_j : X \to Y$, $j \in J$, and a probability distribution Q on J, define $W : X \to Y$ as the channel $W(y|x)=\sum_{j\in J} Q(j)W_j(y|x)$. Then,

$$\sum_{j\in J} Q(j)Z(W_j) \le Z(W). \tag{121}$$

We now write W' as the mixture,

$$W'(y_1^2|u_1)=\frac{1}{2}[W_0(y_1^2|u_1)+W_1(y_1^2|u_1)], \tag{122}$$

where

$$W_0(y_1^2|u_1)=W(y_1|u_1)W(y_2|0), \tag{123}$$
$$W_1(y_1^2|u_1)=W(y_1|u_1\oplus 1)W(y_2|1), \tag{124}$$

and use the above result to obtain the claimed inequality

$$Z(W')\ge\frac{1}{2}[Z(W_0)+Z(W_1)]=Z(W). \tag{125}$$

Also, since $0\le Z(W)\le 1$ and $Z(W'')=Z(W)^2$, we have $Z(W)\ge Z(W'')$. □

Proof of Proposition 4.

By expanding (12), we get

$$\begin{aligned}W(y_1^2|0)W'(y_1^2|1)=&\frac{1}{4}[W(y_1|0)^2+W(y_1|1)^2]W(y_2|0)W(y_2|1)\\&+\frac{1}{4}[W(y_2|0)^2+W(y_2|1)^2]W(y_1|0)W(y_1|1),\end{aligned} \tag{126}$$

and

$$W'(y_1^2|0) - W'(y_1^2|1) = \frac{1}{2}[W(y_1|0) - W(y_1|1)][W(y_2|0) - W(y_2|1)]. \tag{127}$$

Suppose W is a BEC, but W' is not. For the identities given, if there is no pair (y_1, y_2) such that the left sides of the identities are both different from zero, then W' consists only of output pairs (y_1, y_2) with either a uniquely possible input u_1, or with uniformly distributed input over all possible values of u_1. But this is a BEC. So for W' to not be a BEC, then there must exist (y_1, y_2) such that the left sides of (126) and (127) be both different than zero. From (127), and with the previous assumption, we infer that neither y_1 nor y_2 is an erasure symbol for W. But then the RHS of (126) must be zero, which is a contradiction. Thus, W' must be a BEC. From (127), we conclude that y_1^2 is an erasure symbol for W' iff either y_1 or y_2 is an erasure symbol for W. From the union of those two events, we get that the erasure probability of W' is $2\epsilon - \epsilon^2$, where ϵ is the erasure probability of W.

Conversely, suppose W' is a BEC but W is not. Then, there exists y_1 such that $W(y_1|0)W(y_1|1) \neq 0$ and $W(y_1|0) - W(y_1|1) \neq 0$. By taking $y_2 = y_1$, we see that the RHSs of (126) and (127) can both be made nonzero, which contradicts the assumption that W' is a BEC. The proof completes after handling (13) the same way we did with (12). □

Proof of Theorem 2.

Consider the probability space $(\Omega, \Im, P)$. For $\omega \in \Omega$, $i \geq 0$, by Proposition 5, we have $Z_{i+1}(\omega) = Z_i^2(\omega)$ if $B_{i+1}(\omega) = 1$ and $Z_{i+1}(\omega) \leq 2Z_i(\omega) - Z_i(\omega)^2 \leq 2Z_i(\omega)$ if $B_{i+1}(\omega) = 0$. For $\zeta \geq 0$ and $m \geq 0$, define

$$T_m(\zeta) \triangleq \{\omega \in \Omega : Z_i(\omega) \leq \zeta \text{ for all } i \geq m\}. \tag{128}$$

For $\omega \in T_m(\zeta)$ and $i \geq m$, we have

$$\frac{Z_{i+1}(\omega)}{Z_i(\omega)} \leq \begin{cases} 2, & \text{if } B_{i+1}(\omega) = 0, \\ \zeta, & \text{if } B_{i+1}(\omega) = 1, \end{cases} \tag{129}$$

which implies

$$Z_n(\omega) \leq \zeta \cdot 2^{n-m} \cdot \prod_{i=m+1}^{n} (\zeta/2)^{B_i(\omega)}, \ \omega \in T_m(\zeta), n > m. \tag{130}$$

To see why this holds notice that the above recursive relation of $\frac{Z_{i+1}(\omega)}{Z_i(\omega)}$, on each backwards recursion step it adds to the RHS product a factor of 2 if $B_i(\omega) = 0$ and a factor of ζ if $B_i(\omega) = 1$.

For $n > m \geq 0$ and $0 < \eta < 1/2$, define

$$U_{m,n}(\eta) \triangleq \{\omega \in \Omega : \sum_{i=m+1}^{n} B_i(\omega) > (1/2 - \eta)(n - m)\}. \tag{131}$$

Then, for $\omega \in T_m(\zeta) \cap U_{m,n}(\eta)$ we have

$$\begin{aligned} Z_n(\omega) &\leq \zeta \cdot 2^{n-m} \cdot (\zeta/2)^{(1/2-\eta)(n-m)} \\ =>Z_n(\omega) &\leq \zeta \cdot \left[2^{1/2+\eta}\zeta^{1/2-\eta}\right]^{n-m}, \end{aligned} \tag{132}$$

from which, by putting $\zeta_0 \triangleq 2^{-4}$ and $\eta_0 \triangleq 1/20$, we obtain

$$Z_n(\omega) \leq 2^{-4-5(n-m)/4}, \quad \omega \in T_m(\zeta_0) \cap U_{m,n}(\eta_0). \tag{133}$$

Now, we show that (133) occurs with sufficiently high probability. First, we use the following result, which is proved in [1, Lemma 1].

For any fixed $\zeta > 0$, $\delta > 0$, there exists a finite integer $m_0(\zeta, \delta)$ such that

$$P[T_{m_0}(\zeta)] \geq I_0 - \delta/2. \tag{134}$$

Second, we use Chernoff's bound [2, 10, p. 531] to write,

$$P[U_{m,n}(\eta)] \geq 1 - 2^{-(n-m)[1-H(1/2-\eta)]}, \tag{135}$$

where H is the binary entropy function.

Define $n_0(m, \eta, \delta)$ as the smallest n such that the RHS of (135) is greater than or equal to $1 - \delta/2$. It is clear that $n_0(m, \eta, \delta)$ is finite for any $m \geq 0$, $0 < \eta < 1/2$, and $\delta > 0$.

Now, with $m_1 = m_1(\delta) \triangleq m_0(\zeta_0, \delta)$ and $n_1 = n_1(\delta) \triangleq n_0(m_1, \eta_0, \delta)$, we obtain, for $n \geq n_1$,

$$\begin{aligned} P[T_{m_1}(\zeta_0) \cap U_{m_1,n}(\eta_0)] &= P[T_{m_1}(\zeta_0)] + P[U_{m_1,n}(\eta_0)] - P[T_{m_1}(\zeta_0) \cup U_{m_1,n}(\eta_0)] \\ &\geq P[T_{m_1}(\zeta_0)] + P[U_{m_1,n}(\eta_0)] - 1 \\ &\geq I_0 - \delta/2 + 1 - \delta/2 - 1 = I_0 - \delta. \end{aligned} \tag{136}$$

Finally, we tie the above analysis to the claim of Theorem 2. Define$c \triangleq 2^{-4+5m_1/4}$ and

$$V_n \triangleq \{\omega \in \Omega : Z_n(\omega) \leq c2^{-5n/4}\}, \quad n \geq 0, \tag{137}$$

and note that, by (133)

$$T_{m_1}(\zeta_0) \cap U_{m_1,n}(\eta_0) \subset V_n, \ n \geq n_1. \tag{138}$$

So, $P(V_n) \geq I_0 - \delta$ for $n \geq n_1$. On the other hand,

$$P(V_n) = \sum_{\omega_1^n \in X^n} \frac{1}{2^n} 1\left\{Z(W_{\omega_1^n}) \leq c2^{-5n/4}\right\} = \frac{1}{N}|A_N|, \tag{139}$$

where $A_N \triangleq \{i \in \{1, ..., N\} : Z(W_N^{(i)}) \leq cN^{-5/4}\}$, with $N = 2^n$.

We conclude that $|A_N| \geq N(I_0 - \delta)$ for $n \geq n_1(\delta)$. This completes the proof of Theorem 2. □

References

[1] E. Arikan, "Channel Polarization: A Method for Constructing Capacity-Achieving Codes for Symmetric Binary-Input Memoryless Channels", in *IEEE Transactions on Information Theory*, vol. 55, no. 7, pp. 3051-3073, July 2009.

[2] R. G. Gallager, *Information Theory and Reliable Communication*, New York: Wiley, 1968.

[3] K. L. Chung, *A Course in Probability Theory*, 2nd edition, Academic Press, 1974.

[4] E. Arikan and E. Telatar, "On the rate of channel polarization", *2009 IEEE International Symposium on Information Theory*, Seoul, 2009, pp. 1493-1495.

[5] T. Tanaka and R. Mori, "Refined rate of channel polarization", *2010 IEEE International Symposium on Information Theory*, Austin, TX, 2010, pp. 889-893.

[6] W. Wang and L. Li, "Efficient construction of polar codes", *2017 13th International Wireless Communications and Mobile Computing Conference (IWCMC)*, Valencia, 2017, pp. 1594-1598.

[7] I. Tal and A. Vardy, "How to Construct Polar Codes", in *IEEE Transactions on Information Theory*, vol. 59, no. 10, pp. 6562-6582, Oct. 2013.

[8] C. Schürch, "A partial order for the synthesized channels of a polar code", *2016 IEEE International Symposium on Information Theory (ISIT)*, Barcelona, 2016, pp. 220-224.

[9] B. A. Davey and H. A. Priestley, *Introduction to Lattices and Order*, 2nd ed., Cambridge University Press, 2002.