Polar Codes: Primary Concepts and Practical Decoding Algorithms

Kai Niu, Kai Chen, Jiaru Lin, and Q. T. Zhang

ABSTRACT

Polar codes represent an emerging class of error-correcting codes with power to approach the capacity of a discrete memoryless channel. This overview article aims to illustrate its principle, generation and decoding techniques. Unlike the traditional capacity-approaching coding strategy that tries to make codes as random as possible, the polar codes follow a different philosophy, also originated by Shannon, by creating a jointly typical set. Channel polarization, a concept central to polar codes, is intuitively elaborated by a Matthew effect in the digital world, followed by a detailed overview of construction methods for polar encoding. The butterfly structure of polar codes introduces correlation among source bits, justifying the use of the SC algorithm for efficient decoding. The SC decoding technique is investigated from the conceptual and practical viewpoints. State-of-the-art decoding algorithms, such as the BP and some generalized SC decoding, are also explained in a broad framework. Simulation results show that the performance of polar codes concatenated with CRC codes can outperform that of turbo or LDPC codes. Some promising research directions in practical scenarios are also discussed in the end.

INTRODUCTION

The past six decades have witnessed the success of coding theory in digital communications. Claude Shannon's famous Channel Coding Theorem asserts the existence of codes whereby information can be reliably transmitted over a noisy channel at the rate up to the channel capacity. Three essential ideas behind the proof of the channel coding theorem are:

(1) Random code selection.
(2) Joint asymptotic equipartition property (AEP) between the transmitted codeword and the received sequence for large code length.
(3) Optimal maximal likelihood (ML) decoding or suboptimal jointly typical decoding.

The joint AEP plays a central role in the proof, in the sense that it guarantees the received sequence very likely to be jointly typical with the transmitted codeword, and the jointly typical decoding to have a vanishing error probability. Certainly random coding is also important, but only for ease of mathematically proving the existence of a good code.

Approaching capacity with a practical en/decoding complexity is a central challenge in coding theory. Fortunately, during the past two decades a number of "turbo-like" code families, such as turbo codes and low-density parity-check (LDPC) codes, have been found to achieve this goal. The key issue is how to practically implement the ideas used in the proof of the channel coding theorem. Coding randomness is introduced by interleavers in turbo codes or by pseudo-random connections between the variable and check nodes in LDPC codes. For reliable and efficient decoding, turbo codes employ the iterative BCJR (named after its inventors) algorithm, while LDPC codes employ the belief propagation (BP) algorithm. These two algorithms perform only slightly inferior to their ML or maximal a posterior probability (MAP) counterparts.

Given their excellent performance, turbo codes have been successively adopted in 3GPP WCDMA and LTE standards, and so do LDPC codes in IEEE WiMax and 802.11n standards. However, there is no theoretically rigorous proof in the literature to reveal that the transmission over the channel with these codes satisfies the joint AEP. On the contrary, according to the implicative ideas (1) and (3) in the proof of Shannon coding theorem, it has been long believed that the unique approach to devise the optimal capacity-achievable codes is to elaborately combine the pseudo-random encoding and ML/MAP decoding algorithms. The ideas of joint AEP and typical decoding, on the other hand, are only regarded as a method of proof for a long time.

The situation changes due to the recent invention of polar codes by Arikan [1], which opens up a new frontier in the construction of error-correcting codes to achieve the capacity of any given binary input discrete memoryless channel (B-DMC). This new code family is rooted in an elegant effect, called channel polarization, which can be regarded as a Matthew effect[1] in the digital world. At first the same independent channels are transformed into two kinds of synthesized channels with slightly different reliabilities: the good and the bad channels. By recursively applying such polarization transformation over the resulting channels, the reliabilities of the synthesized channels will show significant difference: the "good ones get better

Kai Niu, Kai Chen, and Jiaru Lin are with the Key Lab. of Universal Wireless Communications (Ministry of Education), Beijing University of Posts and Telecommunications.

Q. T. Zhang is with Nanjing University of Posts and Telecommunications.

[1] Matthew effect takes its name from a verse in the biblical Gospel of Matthew. In sociology, this famous effect is used to describe the phenomenon where "the rich get richer and the poor get poorer."

0163-6804/14/$25.00 © 2014 IEEE

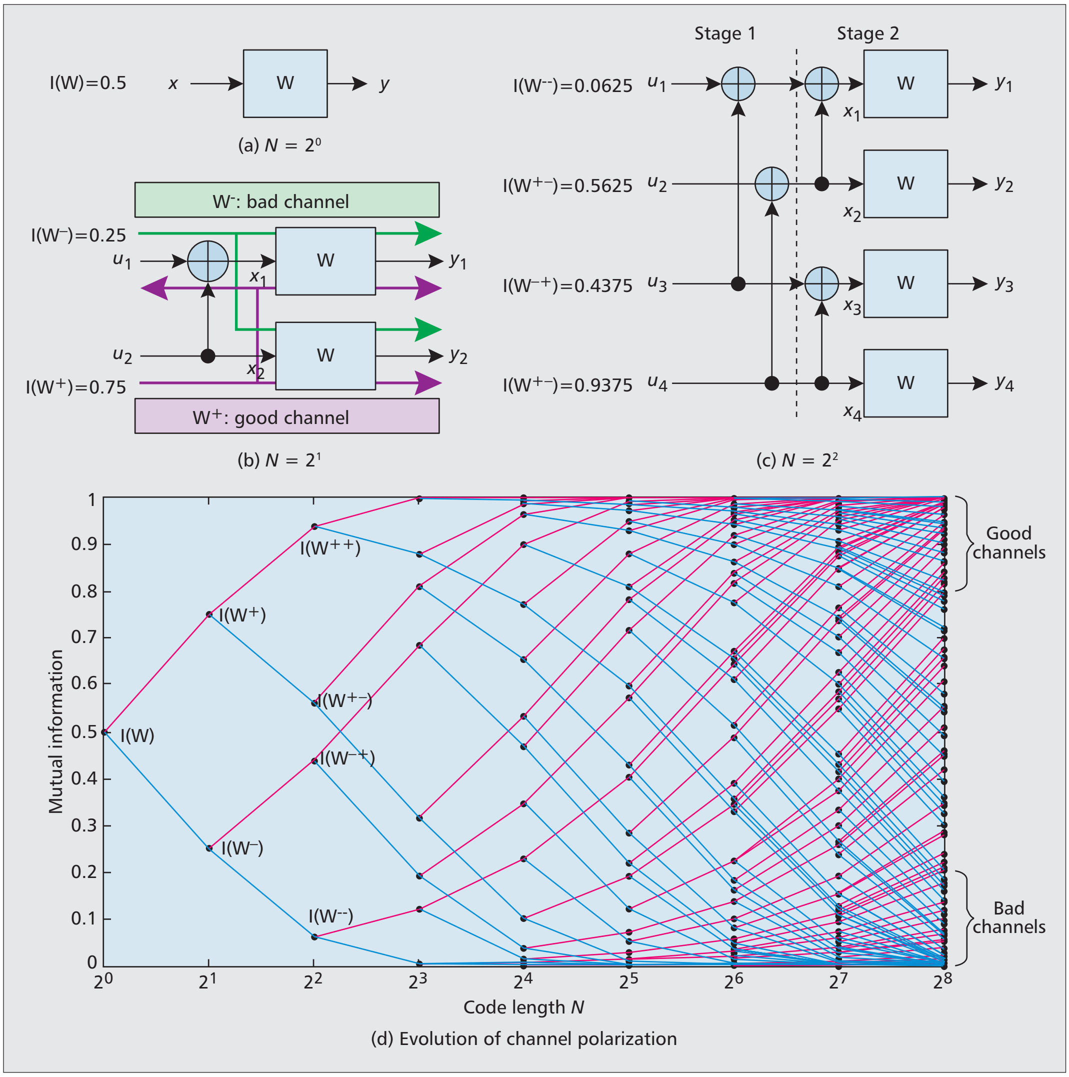


Figure 1. Channel polarization examples in BEC channel with erasure probability 0.5 a) $N = 2^0$; b) $N = 2^1$; c) $N = 2^2$; d) Evolution of channel polarization.

Channel polarization can be recursively implemented by transforming multiple independent uses of a given B-DMC into a set of successive uses of synthesized binary input channels. The polarized channels result from the use of the chain rule to expand the mutual information between the source block and the received sequence.

and the bad get worse." Finally, when the code length is sufficiently large, almost all these synthesized channels trend to two extremes: the noisy channels and those almost free of noise. Therefore, a natural encoding strategy is to transmit free bits (called information bits) over the noiseless channels while assigning fixed bits (called frozen bits) to the noisy ones.

Recall that the joint AEP allows us to divide the set of all pairs of transmitted codeword and received sequence into two sets, the *jointly typical* set, where the sample mutual information is close to the capacity, and the non-joint-typical set, which consists of the remaining pairs with negligible mutual information. As such, the codewords of the jointly typical set can be reliably transmitted over the noiseless channels generated by the channel polarization. Clearly, polar coding is a constructive instance of the joint AEP.

In Arikan's seminal paper [1] a successive cancellation (SC) decoding was proposed as a baseline algorithm which has a very low complexity. By SC decoder for polar codes, we mean a decoder that can estimate and decide bit messages based on the recursive structure of the polar encoder. In principle, SC decoding performs a series of interlaced step-by-step decisions in which a decision in each step heavily depends on the decisions in the previous steps. Due to its susceptibility to error propagation, SC decoding is obviously a suboptimal algorithm. Nevertheless, the error probability of SC decoding can be made arbitrarily small as long as the code length is sufficiently large and the code rate is less than the capacity. Similar to the jointly typical decoding, such an algorithm can asymptotically achieve the optimal performance, without the need of the optimal ML/MAP algorithm or iteration of any form.

Besides the core concept of channel polarization, other techniques critical to the understanding of the polar codes are also included in this article, such as the construction methods, BP decoding, SC decoding and its enhancement algorithms. This tutorial article will guide readers through conceptual explanations about these important issues, aiming at an illustrative exposition from the viewpoint of practical applications. Some further research directions of relevance are also discussed.

The joint typicality, as demonstrated in the behavior of channel polarization, constitutes again a key step to the proof of the channel coding theorem. To the authors' best knowledge, this is probably the first constructive example that achieves the capacity of B-DMC.

CHANNEL POLARIZATION

Channel polarization can be recursively implemented by transforming multiple independent uses of a given B-DMC into a set of successive uses of synthesized binary input channels. The polarized channels result from the use of the chain rule to expand the mutual information between the source block and the received sequence.

As an example of B-DMC, one binary erasure channel (BEC) is given in Fig. 1a, where both the input bit x and output bit y take value 0 or 1. If the erasure probability of this BEC is 0.5, the corresponding capacity is $I(W) = 1 - 0.5 = 0.5$. Combining two independent uses of this BEC, as shown in Fig. 1b, a compound channel (W, W) is obtained which has two input bits x_1, x_2, and two output bits y_1, y_2. Obviously its associated capacity is $2I(W)$. By using one modulo-2 operation $\oplus$ between the two independent BECs, an equivalent compound channel can be obtained which has two input bits u_1, u_2 and the same output bits y_1, y_2. The capacity of this channel is still $2I(W)$. By applying the chain rule of mutual information, this compound channel can be decomposed into two synthesized channels: channel W^- (indicated by the green line with input bit u_1 and the output bits y_1 and y_2), and channel W^+ (indicated by the pink line with input bit u_2 and the output bits u_1, y_1, and y_2). So after channel combining and channel splitting, two independent BECs with the same reliability are transformed into two polarized channels and the sum capacity of two channels is unchangeable, that is, $I(W^-) + I(W^+) = 2I(W)$. The above operation is called single-step transform. In [1] Arikan derived the mutual information of the two channels as $I(W^+) = 2I(W) - I(W)^2$, $I(W^-) = I(W)^2$ and proved that the bad channel W^- has a smaller capacity than the given BEC W, whereas the good channel W^+ has a larger capacity, that is, $I(W^-) \le I(W) \le I(W^+)$. Specifically, for the case of the BEC in this example ($I(W) = 0.5$), the capacities of two polarized channels are $I(W^+) = 0.75$ and $I(W^-) = 0.25$, respectively.

Moreover, for four independent uses of the given BEC, we can continuously apply the single-step transform on the two uses of W^- and W^+, as shown in Fig. 1c. In stage 2, four copies of the BEC W are divided into two groups, and the two BECs in each group are transformed into two polarized channels W^- and W^+. Since the channels belonging to a different group are mutually independent, we obtain two independent copies for each BEC W^- or W^+. In stage 1 the same operations can be applied for these two BECs respectively. Thus channel W^{--} and W^{-+}(W^{+-} and W^{++}) are derived from two uses of channel W^-(W^+). In Arikan's derivation we use channel W^- instead of channel W and obtain the capacities of channel W^{--} and W^{-+}, that is, $I(W^{--}) = I(W^-)^2 = 0.0625$, $I(W^{-+}) = 2I(W^-) - I(W^-)^2 = 0.4375$. Similarly, the capacities of channel W^{+-} and W^{++} can be evaluated as $I(W^{+-}) = I(W^+)^2 = 0.5625$, $I(W^{++}) = 2I(W^+) - I(W^+)^2 = 0.9375$, respectively. Obviously the capacity of W^{--} is further reduced as compared to W^-, whereas W^{++} is further enlarged relatively to W^+. Thus compared with the case of two channels, the polarization effect with four channels becomes more remarkable.

By the same principle the polarizing transformations can be continuously performed over $N = 2^n$ independent uses of the given BEC W and the capacities of all the polarized channels can be recursively evaluated. Figure 1d illustrates the evolution of channel polarization for code length $N = 2^0 \sim 2^8$, where each node (indicated by a black circle) denotes one polarized channel in a certain code length, and the ordinate of the node is the capacity of the corresponding channel. In addition, the line between the adjacent two nodes denotes the evolutional trajectory of the channel polarization. For an example, when code length varies from $N = 2^0$ to $N = 2^1$, one BEC W is evolved into two BECs: W^- and W^+. We use red lines to mark the good channels and blue lines to mark the bad channels, as shown in Fig. 1d. The capacity advantage of the good channel W^+ over the bad channel W^- can be accumulated and the polarization effect continuously improves with increased code length $N = 2^2 \sim 2^8$. This phenomenon is a typical Matthew effect: the good become better and the bad become worse. Observing the evolutional trajectories of good/bad channels, we find that the capacities of most of the polarized channels tend to either 1 (good channels with little noise) or 0 (bad channels with full noise). Equivalently, the error probabilities of the noiseless channels or noisy channels go to 0 or 1.

As the code length N approaches infinity, the polarized channels under the Matthew effect go to two extremes (the corresponding capacity is 0 or 1). Arikan proved the stochastic convergence properties of the capacities of polarized channels by using the Martingale Theory [1]. In particular, for the BEC in this example the proportion of the noiseless channels is exactly equal to the symmetric capacity $I(W) = 0.5$.

Intuitively, transmitting codewords over those noiseless channels is completely error-free. The probability that the transmitted codeword and the received sequence are jointly typical goes to one, as shown by the joint AEP. On the other hand, any randomly chosen codeword and the received sequence are jointly typical only with a probability approximately equal to $2^{-NI(W)}$, which is negligible for a large code length. This allows us to assert that there exists a one-to-one fixed mapping between the transmitted codeword and the received sequence, and that about $2^{NI(W)}$ distinguishable codewords can be reliably transmitted over these channels. The joint typicality, as demonstrated in the behavior of channel polarization, constitutes again a key step to the proof of the channel coding theorem. To the authors' best knowledge, this is probably the first constructive example that achieves the capacity of B-DMC.

ENCODING AND CONSTRUCTION

Since the noiseless channels have higher capacities or lower error probabilities than the noisy channels, the channel polarization phenomenon suggests a new philosophy for channel coding, namely, selecting the noiseless channels for information-bits transmission. Because the code

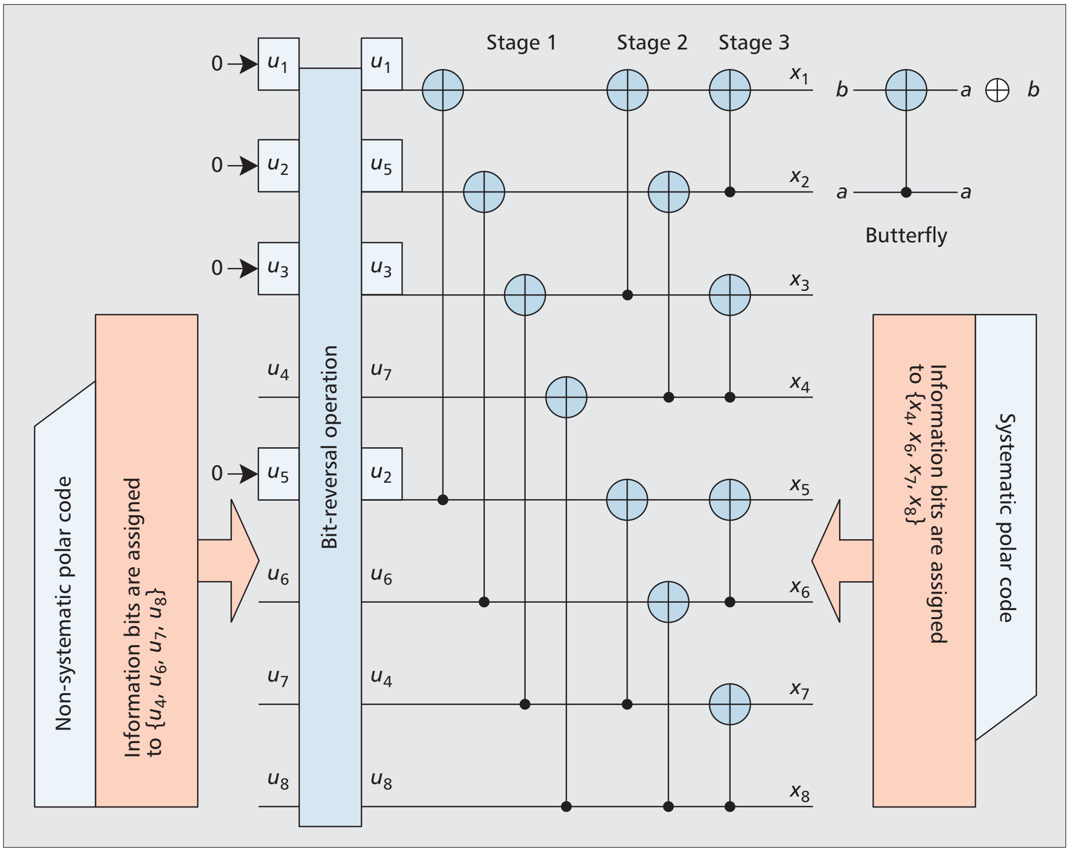


Figure 2. An example of polar coding.

Compared with other coding schemes, this rate-compatible property is a significant advantage. Moreover, unlike the traditional code construction to maximize the minimum Hamming distance, the aim of polar coding is to directly minimize the error probability of the information-bearing polarized channels.

rate can be finely adjusted by adding or deleting one polarized channel, we can almost continuously vary the rate of polar codes. Compared with other coding schemes, this rate-compatible property is a significant advantage. Moreover, unlike the traditional code construction to maximize the minimum Hamming distance, the aim of polar coding is to directly minimize the error probability of the information-bearing polarized channels.

Encoding Principle

Generally, there are mainly three polar encoding schemes: non-systematic coding, systematic coding, and generalized concatenated coding.

In the original polar coding, the non-systematic form is used. Given the code length $N = 2^n$, $n = 1, 2, \ldots$, and information length K, the binary source block $\mathbf{u} = (u_1, u_2, \ldots , u_N)$ consists of K information bits and $N - K$ frozen bits. The codeword $\mathbf{x}$ with code rate $R = K/N$ can be obtained as follows:

$$\mathbf{x} = \mathbf{u}\mathbf{G}_N = \mathbf{F}_2^{\otimes n} \quad (1)$$

where $\mathbf{G}_N = \mathbf{B}_N\mathbf{F}_2^{\otimes n}$ is the generation matrix, $\mathbf{B}_N$ is the bit-reversal permutation matrix, $\mathbf{F}_2^{\otimes n}$ is the n-th Kronecker power of $\mathbf{F}_2$, and

$$\mathbf{F}_2 = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}$$

is called the kernel matrix. Obviously this encoding process is similar to the Walsh-Hadamard transform.

Hereafter, we will use a polar code with $N = 8$, $K = 4$, and $R = 1/2$ as an example. Under this non-systematic scheme, the information bits are assigned to the side of the source block. As shown in Fig. 2, the information bits $\{u_4, u_6, u_7, u_8\}$ are assigned to the polarized channels with the lower error probabilities, while the frozen bits are assigned to the remaining less reliable channels. Each polarized channel is associated with a specific row of the generation matrix. The frozen bits typically take a fixed value of zeros and are assumed known at both the encoder and decoder.

For non-systematic polar codes, the generator matrix corresponding to the information bits can be composed of the rows of $\mathbf{G}_N$ with the lowest error probabilities. In contrast, the Reed-Muller (RM) codes have the similar generation matrix, but the row selection rule is based on the Hamming weight of the rows of G_N. As stated in [1], the RM rule for information bit assignments leads to asymptotically unreliable codes under SC decoding.

In Fig. 2 one butterfly unit can transform two independent input bits (a, b) into two correlated output bits $(a \oplus b, a)$, which is dominated by the kernel matrix $\mathbf{F}_2$ and corresponding to the two-channel polarization. This operation can also be recursively applied to the entire codeword, namely, a codeword $\mathbf{x}$ splits into two parts in stage-3, each of which in turn splits again into two parts in stage-2, and so on, until one reaches single source bit u_i in stage-1. So the process of polar coding for $N = 8$ includes one bit-reversion and three stages of butterfly operations. Generally, given a code length $N = 2^n$, the polarization transform can be decomposed into n

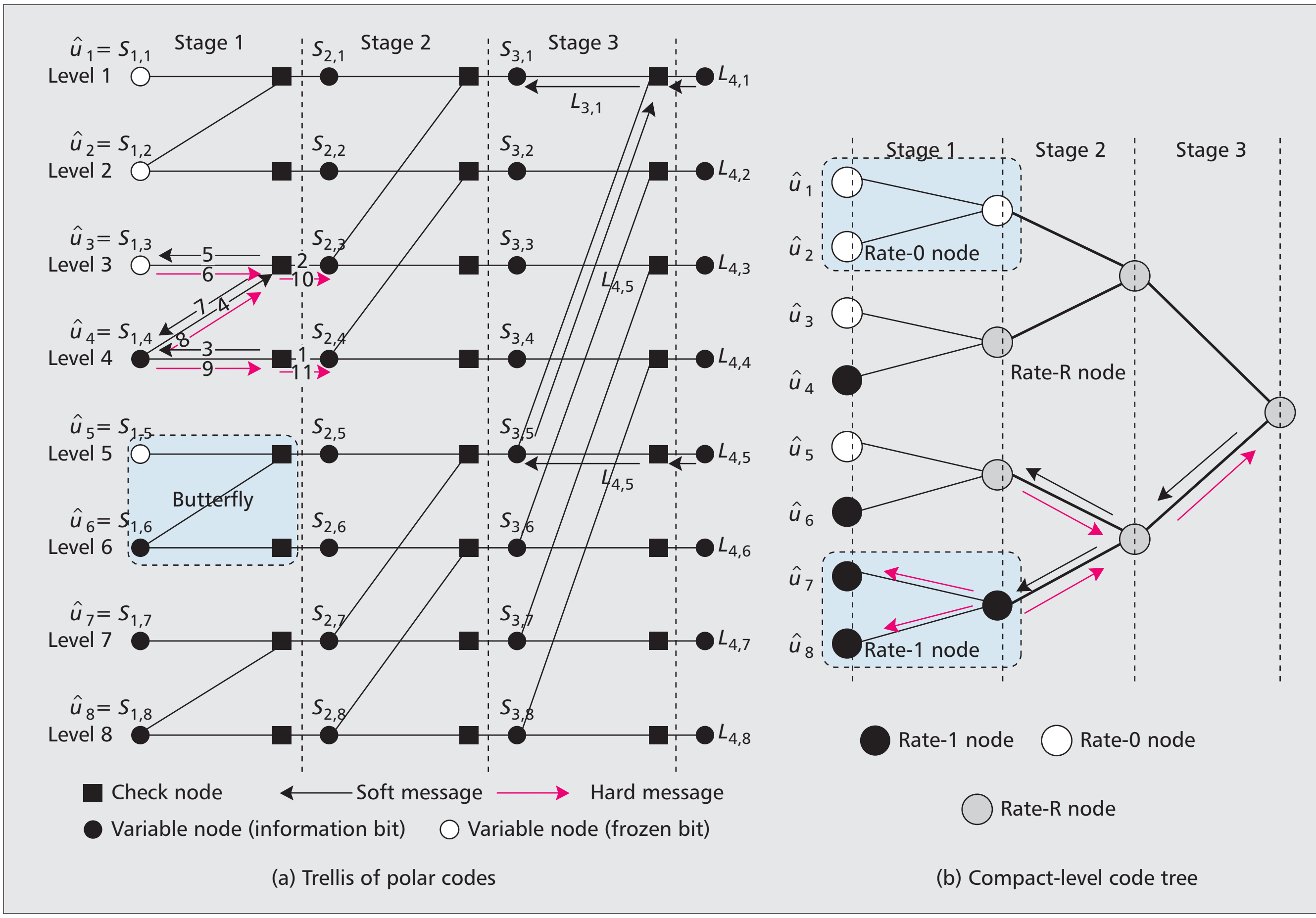


Figure 3. Code tree and trellis representations of polar codes. a) Trellis of polar codes; b) Compact-level code tree.

stages with each of $N/2$ butterfly operations, resulting in an encoding complexity of order $O(N\log N)$.

According to the coding theory, every linear block code can be transformed into an equivalent systematic code. Then a systematic polar coding scheme is introduced in [2]. Unlike the non-systematic scheme, the information bits appear as part of the codeword transparently. With these information bits and the frozen bits at the source block side, the other bits at the codeword side can be determined by some algebraic manipulations. The major advantage of systematic polar codes is that they are more robust against error propagation under SC decoding. Compared with the original (non-systematic) polar code, the systematic polar code has the same block error rate (BLER), but is superior to the former in bit error rate (BER) performance.

When viewed from the framework of generalized concatenated codes, a polar code can be decomposed into a set of outer/inner codes [3]. Namely, in the recursive structure of polar codes, the first l stages can be considered as 2^{n-l} outer codes with length $N_0 = 2^l$, and the rest $n - l$ stages as inner codes with length $N_i = 2^{n-l}$. These outer/inner codes are all polar codes with smaller code lengths. Since the block-wise ML decoding is used for the outer codes and SC decoding is used for the inner codes, the error propagation is constrained in the interior of the inner codes. Compared with the SC decoding over the entire codeword, this decoding strategy is beneficial to the BLER performance.

Channel Selection

The BLER performance of polar codes with SC decoding is upper-bounded by the summation of the error probabilities of all the information-bearing polarized channels [1]. Thus an important step to construct a polar code with a code length N and information bit length K is, in essence, to select the K most reliable channels such that the BLER upper bound is minimized. Therefore reliability calculation and channel selection are two central issues to the code construction.

In his seminal paper [1] Arikan proposed a recursive calculation algorithm based on Bhattacharyya parameters for channel-reliability evaluation. If the original B-DMC is a BEC, the erasure probabilities of the polarized channels can be tracked with a low complexity of $O(N \log N)$ by using this recursive algorithm. Equivalently, the corresponding capacities can also be recursively calculated for the case of BEC, as shown in Fig. 1. The complexity can be further reduced to $O(N)$ if the intermediate Bhattacharyya parameters are used instead [4]. However, for channels other than BEC, the computational complexity grows exponentially with the code length and input alphabet size.

To construct a polar code over an arbitrary symmetric B-DMC, Mori and Tanaka [4] proposed the use of density evolution (DE) tools for tracing the probability density function (PDF) of log-likelihood ratios (LLRs) at the variable and check nodes in the decoding graph of polar codes (e.g. Fig. 3). The convolutions of the LLR PDFs are performed at the variable and check nodes. This DE technique is widely used in designing LDPC codes, and is equally applicable to the polar codes design. Based on the LLR PDFs at the first stage of the variable nodes, the error probabilities of all the polarized channels can be obtained. Using the same arguments as the case with Bhattacharyya parameters, the order of magnitude of convolutions is also $O(N)$. In practical implementation, in order to keep the complexity to an acceptable level, the LLR PDFs should be quantized into q levels. Thus, the computational complexity of the quantized density evolution is $O(q^2N)$. However, a typical value of q is 10^5, implying a huge computational burden in practical application. The difficulty is further aggravated by the quantization errors, which are accumulated over multiple polarization stages. Indeed, as noted in [4] it is difficult to find optimal tradeoff between the implementation complexity and calculation precision.

Tal and Vardy [5] proposed an effective method to solve this problem by controlling the quantization errors and through appropriate approximation. They wisely introduced two approximation methods, called upgrading and degrading quantization, to get a lower and upper bound on the error probability of each channel. Both methods transform the relevant channel into a new one with a smaller output alphabet in terms of μ. Then the construction complexity with their algorithm can be evaluated as $O(N\mu^2\log\mu)$. A typical value of μ is 256, so it is much less complex than DE.

Although the precision of density functions of this algorithm can be improved by increasing output alphabet size, the algorithm complexity also increases. For binary input additive white Gaussian noise (AWGN) channels, mostly concerned by coding theorists, an alternative method [3] called Gaussian approximation (GA) can be applied in the construction of polar codes. The GA has a lower complexity than Tal and Vardy's method when applied to binary input AWGN channels, but yielding almost the same precision. From the practical viewpoint GA is a more attractive choice than other methods.

SUCCESSIVE CANCELLATION DECODING

Recall that the butterfly units in the polar encoder introduce the correlation between the source bits, so that each coded bit with a given index relies on all of its preceding bits with smaller indices. This kind of correlation can be conceptually treated as interference in the source-bit domain which, when exploited, leads to a much better decoding performance and thus constitutes the central idea of a basic decoding algorithm, called SC decoding. Successive cancellation of the "interference" caused by the previous bits improves the reliability in the retrieval of source bits. Due to the regular structure of polar codes, the SC algorithm can be described in terms of a trellis or a code tree structure.

$$L_{i,j}=\begin{cases}2\tanh^{-1}\left[\tanh\left(\dfrac{L_{i+1,j}}{2}\right)\cdot\tanh\left(\dfrac{L_{i+1,j+2^{i-1}}}{2}\right)\right], & \left\lfloor\dfrac{j-1}{2^{i-1}}\right\rfloor \bmod 2=0\\ \left(1-2s_{i,j-2^{i-1}}\right)\left(L_{i+1,j-2^{i-1}}\right)+L_{i+1,j}, & \text{otherwise}\end{cases} \quad (2)$$

DECODING PRINCIPLE

SC decoding can be viewed as a soft/hard message passing algorithm over the trellis of polar code. The trellis consists of n stages and N levels. Each stage includes $N/2$ butterfly units and each unit contains a pair of check and variable node. SC decoder updates the messages stage by stage and sequentially decides the estimation bit by bit.

Figure 3a shows the trellis structure for code length $N = 8$. The hard messages propagated in the trellis are the estimation bits corresponding to the variable nodes designated as $s_{i,j}$, where i and j indicate the stage and level index in the trellis, respectively, $1 \le i \le n + 1$, $1 \le j \le N$. The soft messages corresponding to these bits are the LLR values denoted as $L_{i,j} = L(s_{i,j})$. In this example the bit values in the left side of the trellis are the estimations of the source block, that is, $s_{1,j} = \hat{u}_j$, and the corresponding soft messages are $L_{1,j} = L(\hat{u}_j)$. Let y_j be the received signal after bit-reversal permutation, the corresponding LLR is denoted by

$$L_{n+1,j}=\log\frac{P(y_j\mid 1)}{P(y_j\mid 0)}.$$

The update and decision rule can be expressed as follows.

Soft Messages Updated Rule — See Eq. 2 above where, $i = 1, 2, \ldots, n$, $j = 1, 2, \ldots, N$, $\tanh(\cdot)$ is the hyperbolic tangent function, and $\lfloor\cdot\rfloor$ is the floor function.

In words, the soft message at the check node with level index satisfying $\lfloor(j-1)/2^{i-1}\rfloor \bmod 2 = 0$ is updated by using the first formula of Eq. 2, which is similar to the computation at the check node of LDPC codes; while the message updating at the variable node is performed by the second one of the equation, which requires the knowledge of the variable node constraints and the hard message.

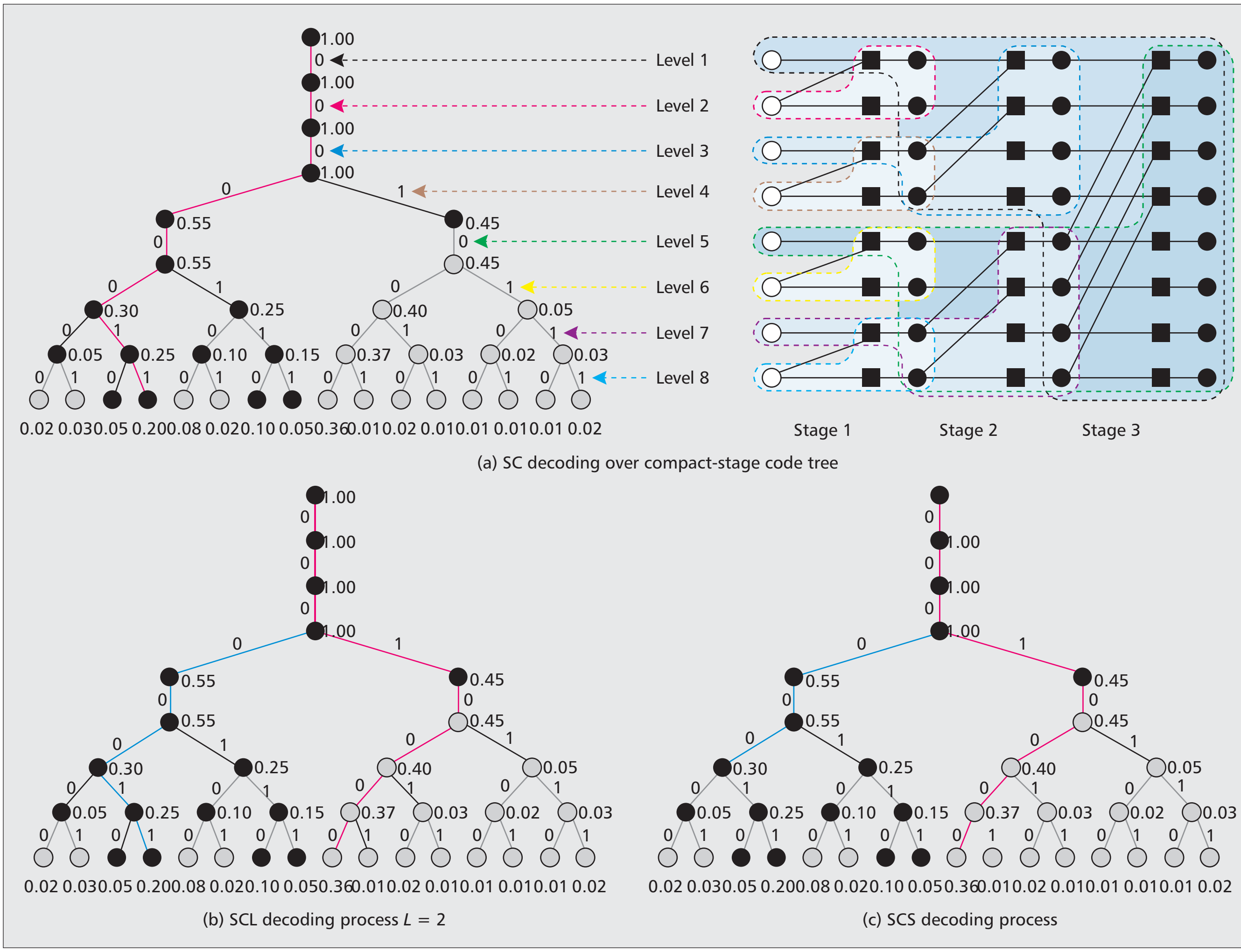
Hard Messages Updated Rule —

$$S_{i+1,j}=\begin{cases}s_{i,j}\otimes s_{i,j+2^{i-1}}, & \left\lfloor\frac{j-1}{2^{i-1}}\right\rfloor \bmod 2=0\\ s_{i,j}, & \text{otherwise}\end{cases} \quad (3)$$

where $\oplus$ is the modulo-2 operation and the other notations are the same with Eq. 2.

In words, the hard message at the check node with specific level index is updated by using the first formula of Eq. 3; otherwise, the hard message is updated by the second formula.

Decision Rule — In stage 1 the bit decision rule is as follows: for an information bit, if soft messages $L_{1,i} \ge 0$, then $\hat{u}_i = 0$, otherwise $\hat{u}_i = 1$; for a frozen bit, simply set it to a pre-determined value, for example, $\hat{u}_i = 0$.



(a) SC decoding over compact-stage code tree

(b) SCL decoding process $L = 2$

(c) SCS decoding process

Figure 4. SC, SCL and SCS over compact-stage code tree. a) SC decoding over compact-stage code tree; b) SCL decoding process L = 2; c) SCS decoding process.

In the trellis example shown in Fig. 3a, the SC decoder performs the soft message calculations and passes through the trellis from right to left; then, the hard messages are calculated and propagated in the reverse direction. As an instance, in the first butterfly unit (from the top of the trellis) of the stage-3, soft message $L_{3,1}$ is computed by using $L_{4,1}$ and $L_{4,5}$. Further, as another instance, in the second butterfly unit (from the top of the trellis) of the stage-1, soft message $L_{2,4}$ and $L_{2,3}$ are sent to the related check nodes after step 1 and 2 (illustrated by the arrows and numbers), respectively. Then after steps 3~6, the soft message $L_{1,3}$ is obtained by Eq. 2. However, no matter what value $L_{1,3}$ takes, the corresponding hard message $s_{1,3} = 0$ is sent back to the check node because bit u_3 is a frozen bit. Then the soft message $L_{1,4}$ is calculated in step 7 by using Eq. 2 with the hard message ($s_{1,3}$) and soft messages ($L_{2,4}$ and $L_{2,3}$). The corresponding hard message $s_{1,4}$ is sent back to the check nodes in steps 8 and 9, respectively. In steps 10 and 11, the hard message $s_{2,3}$ and $s_{2,4}$ are calculated by using Eq. 3.

A direct implementation of the SC decoder is to activate the butterfly units one by one. But this sequential scheduling has poor throughput performance. By simultaneously activating multiple butterfly units, a higher processing throughput can be achieved; that is widely concerned in the hardware design.

The soft message calculations in one butterfly unit can be counted as a basic operation in SC decoding. Since the trellis consists of $(N/2)\log_2 N$ butterfly units, the overall time complexity of the SC decoder is $O(N\log N)$ and it requires $O(N\log N)$ memory units to store the LLR values.

Simplified SC Decoding

From a practical viewpoint, reducing the decoding complexity is an important issue. A simplified successive-cancellation (SSC) decoder [6] was proposed to decrease the redundant calculations in SC decoding without affecting the error performance. According to the feature of the progressive polarization, polar codes can be presented by a compact-level code tree, as shown in Fig. 3b. In the third stage only two levels are kept because all the channels are polarized into two kinds. Similarly, there are four and eight levels in the other two stages, respectively. All the nodes in the code tree can be divided into three types: rate-zero nodes, rate-one nodes, and rate-

R nodes, whose descendants are all frozen bits, all information bits, and partial frozen and information bits, respectively.

The operations in the SSC decoding can be summarized as follows: for constituent codes consisting of rate-one nodes or rate-zero nodes, hard messages are passed directly or no operations are performed; otherwise, for rate-R nodes, standard SC decoding operations are processed. Compared with the original SC decoding, the complexity is reduced about 2~20 times by the simplification in rate-zero and rate-one nodes.

More Powerful Decoding Algorithms

Although polar codes achieve the capacity asymptotically, their performance with SC decoding is unsatisfactory when the code has a finite length. Several alternative decoding schemes have been proposed to improve the finite-length performance of polar codes, such as successive cancellation list/stack (SCL/SCS) and BP decoding algorithms.

Improved SC Decoding

Compact-Stage Code Tree — We can use a unified framework, called compact-stage code tree, to describe SC decoding and its improved algorithms, such as SCL/SCS decoding. Figure 4a gives an example of SC decoding over a compact-stage code tree, which consists of eight levels, with each level related to a piece of trellis which is circled by a color-dashed box .

In this code tree, except for the leaf nodes and the frozen nodes, each node has two descendants and the corresponding branches are labeled with 0 and 1, respectively. A decoding path consists of a branch sequence from the root to one of the nodes and the corresponding reliability can be measured using a posteriori probability (APP). In Fig. 4 the number written next to each node provides the APP metric of the decoding path from the root to that node. The black circles represent the nodes that are visited (the APP metrics of which are calculated), and the gray ones are those that are not visited in the search process.

Recall that $s_{i,j}$ is the bit estimation corresponding to the variable node in stage-i and level-j, which is covered by some pieces of the trellis and is related to a certain decoding path. Similar to Eq. 2, the APP metrics can be calculated in a recursive manner as well. From the practical view, the logarithmic APP metrics have better numeric stability than other expressions, the details of which are interpreted in [7].

SCL and SCS — The SC decoding of polar codes can be regarded as a greedy search algorithm over the compact-stage code tree. Between the two branches associated with an information bit at a certain level, only the one with the larger probability is selected for further processing. Whenever a bit is wrongly determined, correcting it in the future decoding procedure becomes impossible. The red bold branches in Fig. 4a illustrate an SC decoding path "00000011." Obviously this decoding path is not optimal due to the level by level decision strategy.

As an enhanced version of SC, the SCL decoder proposed in [8] searches the code tree level by level, in much the same manner as does the SC. However, unlike SC where only one path is reserved after processing at each level, the SCL can be regarded as a breadth-first search algorithm and allows a maximum of L candidate paths to be further explored. At each level related to an information bit, SCL doubles the number of candidates by appending a bit 0 or 1 to each of the candidate paths. It then selects a maximum of L ones with the largest metrics and stores them in a list. Figure 4b shows an example of SCL with list size $L = 2$. In each level, two candidate paths (illustrated by blue and red bold edges) are reserved, and the most reliable path "000100000" is found.

The SCS decoder proposed in [9] uses an ordered stack to store the candidate paths and tries to find the optimal estimation by searching along the best candidate in the stack. Whenever the top path in the stack that has the largest path metric reaches a leaf node, the decoding process stops and outputs this path. Unlike the candidate paths in the list of SCL, which always have the same length, the candidates in the stack of SCS have different length. Figure 4c gives a simple example of SCS decoding. Compared with the SCL decoding, SCS can also find the same optimal path but the two candidate paths have different lengths so the number of path extensions can be reduced.

As for the implementation aspect, a space-efficient structure is suggested in [8] to implement the SC decoder, and the time and space complexities are $O(N\log N)$ and $O(N)$, respectively. A direct implementation of the SCL decoder will require $O(LN^2)$ computations. In [8] a so-called "lazy copy" technique based on the memory sharing structure among the candidate paths is introduced to reduce the redundant copy operations. Therefore the SCL decoder can be implemented with time complexity $O(LN\log N)$. Similar to that for the SC and SCL decoders, these features can also be applied in the implementation of the SCS decoders. The actual computations of SCS, supposing the depth of stack D is large enough, become much fewer than that of SCL when working in the moderate or high signal-to-noise ratio (SNR) regime.

Hybrid Decoding — Combining the principles of SCL and SCS, a new decoding algorithm called successive cancellation hybrid (SCH) is proposed in [10]. This proposed algorithm can provide a flexible configuration when the time and space complexities are limited. Under proper configurations, all three improved SC decoding algorithms, such as SCL, SCS, and SCH, can approach the performance of ML decoding but with acceptable complexity. With the help of the proposed pruning technique, the time and space complexities of these decoders can be significantly reduced and be made very close to those of the SC decoder in the high SNR regime.

CRC-Aided Decoding

To further improve the performance of polar codes, CRC (cyclic redundancy check)-aided SCL/SCS decoding schemes, such as CA-

In this code tree, except for the leaf nodes and the frozen nodes, each node has two descendants and the corresponding branches are labeled with 0 and 1, respectively. A decoding path consists of a branch sequence from the root to one of the nodes and the corresponding reliability can be measured using a posteriori probability (APP).

SCL/CA-SCS, have been proposed recently in [7]. In these schemes the SCL/SCS decoder outputs the candidate paths into a CRC detector, and the check results are utilized to detect the correct codeword. To lower the time complexity of SCL decoding brought by a large list size, an adaptive CRC-aided SCL decoder (aCA-SCL) is proposed in [11] by progressively enlarging the list size. Under these CRC-aided decoding schemes, the performance of polar codes is substantially improved and outperforms that under ML decoding.

	Encoding		Design and construction	
	Structure	Complexity	Methods	Complexity
Polar	Recursive encoder [1]	$O(N\log N)$ medium	DE [4]	High
			Tal and Vardy [5]	Medium
			GA[3]	Low
Turbo	Convolutional encoder [12]	$O(mN)$ low	Interleaver optimization [12]	High
LDPC	Matrix multiplication [12]	$O(N^2)$ high	Degree distribution optimization [12]	High
Decoding				
	Algorithms	Complexity	Performance	
Polar	SC[1]	$O(N\log N)$ low	Suboptimal	
	SCL [8]	$O(LN\log N)$ medium	Approach ML	
	BP[1]	$O(I_{max}N\log N)$ high	Suboptimal	
	CA-SCL[7]	$O(LN\log N)$ medium	Outperform ML	
Turbo	Iterative BCJR	$O(I_{max}(4N2^m))$ high	Approach ML	
LDPC	BP	$O(I_{max}(N\bar{d}_v + M\bar{d}_c))$ High	Approach ML	
Notation list	The turbo code contains two convolutional component codes m: the memory length of component code I_{max}: the maximum iteration number N: code length $M = N - K$: the number of check nodes $\bar{d}_v(\bar{d}_c)$: the average of variable (check) degree distribution of the LDPC code			

Table 1. A summary of comparisons of polar, turbo and LDPC codes.

Belief Propagation Decoding

Based on the factor graph representation which is equivalent to the trellis of polar code shown in Fig. 3a, the BP decoder of polar codes is first introduced by Arikan in [1]. Rather than exchanging the hard messages, soft messages are transferred between the check and variable nodes in the BP decoding. Thus the BP decoder can considerably outperform the SC decoder. However, the message-passing schedule in BP plays an important role for channels other than BECs due to the redundant loops on the factor graph, and the optimal schedule is hard to determine. Besides, it is difficult to improve the parallelism of the BP decoder. Therefore, a practical BP decoder suffers a higher implementation complexity and offers a lower throughput than an SC decoder.

ML or MAP Decoding

Theoretically the optimal decoding algorithms of polar codes are ML or MAP decoders, which can be implemented via Viterbi and BCJR algorithms on the trellis, respectively. But these well known algorithms are too complex to be practical for a medium or long code length, so they are only regarded as a reference of performance comparison for other decoding algorithms of polar codes.

Pros and Cons

Polar codes, coined by Arikan, provide a novel idea for the capacity-approaching code construction and create a new field of coding theory. Although the polar codes have perfect theoretic results, compared with the other advanced channel coding schemes, such as turbo or LDPC codes, there still exist many open issues for practical applications. A summary of comparisons for these three codes is shown in Table 1. We directly quote some properties or conclusions on the turbo or LDPC codes from the book [12]. The reader interested in the details can consult [12] and the references therein.

Advantages of Polar Codes

Concerning the implementation of encoding, the encoder of polar codes utilizes a recursive structure with a medium complexity $O(N\log N)$. Moreover, given the code length $N = 2n$, the code rate of polar codes can be finely tuned by adding or deleting one polarized channel. Since the turbo encoder consists of two convolutional component codes, the corresponding complexity is linear in the code length N. On the contrary, for irregular LDPC codes based on the random construction, the encoding complexity $O(N^2)$ is very high. Fortunately this complexity can be efficiently reduced based on a greedy upper-triangulation process [12]. In practice an LDPC code is often constrained further to yield a simple encoding, such as quasi-cyclic (QC)-LDPC code. So all three codes have efficient encoder structures, but polar codes have a better rate-compatible property to almost continuously vary the code rate.

With respect to the code design and construction, polar codes can be constructed with different methods, such as DE [4], Tal and Vardy [5], or GA [3] algorithms. In practice the GA algorithm is a preferable choice since it can provide sufficiently precise evaluation of error performance with the lowest complexity. On the other hand, in order to optimize turbo codes we use the extrinsic information transform (EXIT) [12] as a semi-analytic tool to predict error performance. In addition, as a core building block the interleaver in the turbo encoder should be optimized based on computer searching. Clearly the design of turbo codes is more complex than that of polar codes. In the same manner the optimization of LDPC codes is also a time consuming process. The variable and check degree distribution can only be optimized by computer searching [12]. And the primary method to generate the parity-check matrix (or factor graph) of LDPC code is also a computer-aided construction process, such as the progressive edge growth (PEG) algorithm [12]. In a word, the construction of polar codes is more straightforward and simpler than the other codes.

Theoretically all three codes have similar asymptotic performance when the code length tends to be infinite. The polar codes based on SC decoding have been proved to achieve the capacity of symmetric B-DMC because the code construction accords with the joint AEP property. On the contrary, the turbo/LDPC codes based on BCJR/BP decoding show the tendency of approaching the capacity by the computer simulation rather than the theoretic proof. Thus polar codes have a theoretic advantage over the other two codes.

Nevertheless, for a finite code length all three codes can be uniformly presented by the associated factor graph and perform similar message-passing algorithms on the graph. For the iterative decoding of turbo codes, each component decoder performs the BCJR algorithm and exchanges the extrinsic information via the interleaver/de-interleaver. For LDPC codes, the BP algorithm runs on the factor graph and soft messages are passed between the variable and check nodes. After many iterations both decoders can work well and approach the performance of ML decoding. However, for polar codes the performance of the BP algorithm is limited by the short-length loops of the factor graph and inferior to that of ML decoding. Therefore some improvements of SC in the code tree are better choices for polar codes, such as SCL/SCS/CA-SCL/CA-SCS and so on.

Figure 5 gives the BLER performance comparisons of various coding and decoding schemes. All the coding schemes have the code length $N = 1024$ (except for the LDPC code, which has $N = 1056$) and the code rate $R = 1/2$, and are transmitted over binary input AWGN channels. In addition, 24 CRC parity bits are attached to every input block of the encoders. For turbo codes the two encoding schemes are respectively referred to as the WCDMA and LTE standards, and the Log-MAP decoding algorithm (BCJR decoding in logarithmic domain) with a maximum of eight iterations ($I_{\max} = 8$) is applied. Similarly, the LDPC codes are constructed according to the WiMax standard, and a standard BP algorithm with a maximum of 200 iterations ($I_{\max} = 200$) is utilized.

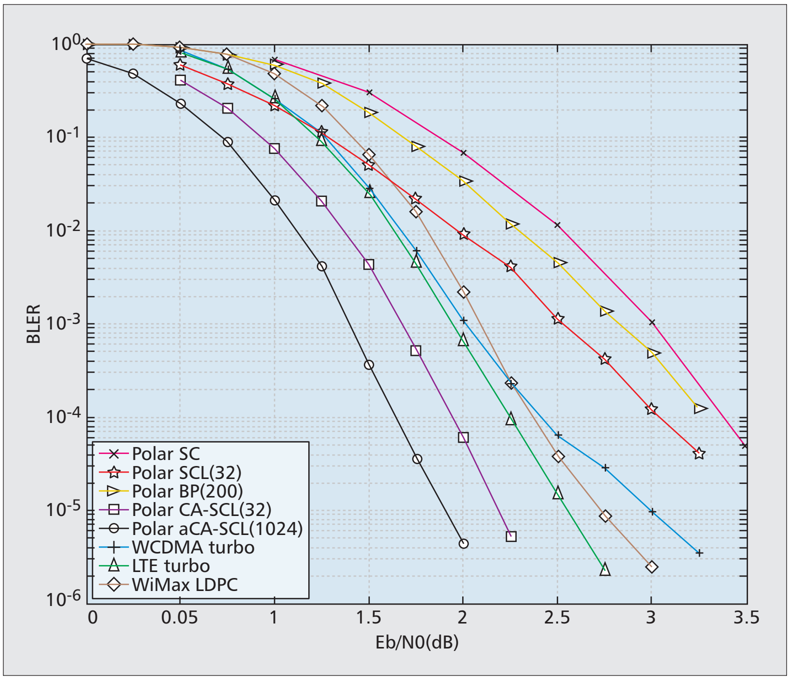


Figure 5. Performance comparison of polar, turbo and LDPC codes.

When constructing the polar codes, the polarized channels carrying information bits are selected using the method in [5]. The list size of CA-SCL is $L = 32$ and the maximum list size of aCA-SCL is $L = 1024$. The number of maximum iterations of the BP decoder of polar codes is $I_{\max} = 200$.

We can see that the BLER of SCL/BP is better than that of SC. But there still is about a 0.7 dB performance gap between the turbo/LDPC codes and polar codes at the BLER of 10^{-4}. Fortunately, with a sufficient big list polar codes under the CRC aided decoding algorithms can outperform the turbo/LDPC codes by up to 0.5 dB~1dB at the BLER of 10^{-4}. Moreover, all the polar coding schemes show no sign of error floors in the simulated SNR regime, which is another advantage over the turbo/LDPC codes.

In order to evaluate the complexity of all the decoding algorithms of these codes, we can use the LLR, logarithmic likelihood probability, or APP calculation as a basic unit. Thus the SC decoder has the lowest complexity $O(N\log N)$, but the corresponding performance is poor. Similarly, turbo decoder executes $(4N2^m)$ basic operations for one iteration, which accounts for 2^m trellis states and four metric calculations per trellis node. Assuming the average of variable (check) degree distribution of LDPC code is $\bar{d}_v(\bar{d}_c)$ and the factor graph contains N variable nodes and $M = N - K$ check nodes, the BP decoder performs $(N\bar{d}_v + M\bar{d}_c)$ basic operations for one iteration. Generally, for an LDPC code with structure constraints, for example, QC-LDPC, the complexity of the BP decoder is slightly lower than that of the turbo decoder. However, the complexities of both algorithms are higher than that of the CA-SCL/CA-SCS decoder of polar codes with proper parameters. The detailed comparisons can be found in [7, 10]. Hence, polar codes with CA-SCL/CA-SCS

Since SC and its improvement algorithms are performed by a level-by-level decoding order in the code tree, these decoders essentially possess a serial architecture. Due to this structure constraint, the decoder design with high throughput has become a challenge by now, which is a main disadvantage of polar codes in practice.

can achieve better tradeoff between performance and complexity than the others.

In conclusion we find that polar codes have evident benefits on the theoretic performance, construction methods, and encoding/decoding schemes, when compared to the turbo/LDPC codes in many aspects.

Disadvantages of Polar Codes

Of course, polar codes have a few disadvantages because there is no free lunch. Recall that, for channel selection of polar codes, the channel reliabilities should be calculated according to a specific channel type or channel condition. Hence, code construction is channel dependent, although the encoder structure is universal. For example, given the binary input AWGN channel, the channels selected to carry information bits should be optimized point-by-point in the given SNR range in order to achieve the optimal performance at each SNR. But this is not a serious problem in practice. Because the error performance of polar codes is not sensible on slight change of information channel set, we can optimize the channel selection at a fixed SNR. So the channel set is uniform within a wide range of SNR yet the performance may have a little loss.

Since SC and its improvement algorithms are performed by a level-by-level decoding order in the code tree, these decoders essentially possess a serial architecture. Due to this structure constraint, the decoder design with high throughput has become a challenge by now, which is a main disadvantage of polar codes in practice.

Future Research Directions

As a popular phenomenon, many building blocks or application scenarios of digital communication systems have the polarization effect, as pointed out by Arikan and many other researchers, such as source coding, channel coding, modulation, relay transmission, multiple-access, and so on. We mainly discuss some open questions of three directions from the practical viewpoint.

First, the most important issue is to design a polar decoder with high throughput. Reducing the calculation operations or increasing the parallelism of the decoder are two efficient methods to decrease the decoding latency, or equivalently, to increase the throughput. For the former, recall that SSC [6] decoding can cut down the unnecessary calculations of the Rate-0 and Rate-1 nodes, therefore it is a typical solution to improve the throughput of the SC decoder. Similarly, the SCS [9] or SCH [10] decoder is a low complexity substitute for SCL. For the latter, Leroux *et al.* proposed a pipeline tree and some enhanced structures to improve the parallel efficiency of the SC decoder [13]. However, for the improved SC decoders, it still is an open problem to design a practical scheme to achieve the same throughput as that of the turbo/LDPC decoders.

Second, the joint design of coding and modulation based on polar codes is an important direction to approach the capacity of the AWGN channel in the high SNR regime. Recently, Seidl *et al.* proposed a uniform framework to jointly describe and optimize both polar coding and modulation [14]. They regarded the pulse-amplitude modulation (PAM) as a general channel polarization phenomenon and suggested two design schemes. One is polar-coded modulation based on the multilevel coding (MLC) approach; the other is a bit-interleaved scheme based on bit-interleaved coded modulation (BICM) technique. These two promising schemes can achieve good tradeoff between power efficiency and spectral efficiency. There are several open problems under this framework, such as the optimization of labeling rule, the low-complexity multi-stage decoding, the universal construction method applying for AWGN and fading channels, and so on.

Finally, the hybrid automatic repeat request (HARQ) schemes based on polar codes will be a key technique in the link adaptive transmission, especially for future wireless communications. Considering the code length of the original polar coding is limited to a power of two, Chen *et al.* proposed a novel class of rate-compatible polar (RCP) codes to increase coding flexibility [15]. The RCP codes are constructed by punctured and repetitive operations on the original polar codes. Like the incremental redundancy (IR)-HARQ scheme, the proposed HARQ scheme based on RCP codes can achieve the same performance as the existing schemes based on turbo/LDPC codes by using a simple SC decoding. More flexible HARQ schemes and more powerful decoding algorithms will be valuable subjects deserving of further exploration.

Conclusions

In this article the primary concepts of polar coding are reviewed, which include channel polarization, the encoding scheme, the construction methods, and various decoding algorithms such as ML, BP, SC, and the improved versions of SC. The coding theory and design methodology of polar codes have been substantially established, while polar codes under CRC-aided decoding can significantly outperform the turbo/LDPC codes. Although there remain some open issues, it is believed that polar coding will be a competitive channel coding solution in future communication systems.

Acknowledgments

The author wishes to thank the anonymous reviewers for their many insightful comments. This material was supported in part by the National Natural Science Foundation of China (no. 61171099), the National Science and Technology Major Project of China (no. 2013ZX03003-012), and Qualcomm Corporation.

References

[1] E. Arikan, "Channel Polarization: A Method for Constructing Capacity-Achieving Codes for Symmetric Binary-Input Memoryless Channels," *IEEE Trans. Info. Theory*, vol. 55, no. 7, Jul. 2009, pp. 3051–73.

[2] E. Arikan, "Systematic Polar Coding," *IEEE Commun. Lett.*, vol. 15, no. 8, Aug. 2011, pp. 860–62.

[3] P. Trifonov, "Efficient Design and Decoding of Polar Codes," *IEEE Trans. Commun.*, vol. 60, no. 11, Nov. 2012, pp. 3221–27.

[4] R. Mori and T. Tanaka, "Performance of Polar Codes with the Construction Using Density Evolution," *IEEE Commun. Lett.*, vol. 13, no. 7, Jul. 2009, pp. 519–21.

[5] I. Tal and A. Vardy, "How to Construct Polar Codes," *IEEE Trans. Info. Theory*, vol. 59, no. 10, Oct. 2013, pp. 6562–82.
[6] A. A. Yazdi and F. R. Kschischang, "A Simplified Successive-Cancellation Decoder for Polar Codes," *IEEE Commun. Lett.*, vol. 15, no. 12, Dec. 2011, pp. 1378–80.
[7] K. Niu and K. Chen, "CRC-Aided Decoding of Polar Codes," *IEEE Commun. Lett*., vol. 16, no. 10, Oct. 2012, pp. 1668–71.
[8] I. Tal and A. Vardy, "List Decoding of Polar Codes," *IEEE Int'l. Symp. Info. Theory (ISIT)*, 2011, pp. 1–5.
[9] K. Niu and K. Chen, "Stack Decoding of Polar Codes," *Electron. Lett.*, vol. 48, no. 12, June 2012, pp. 695–96.
[10] K. Chen, K. Niu, and J. Lin, "Improved Successive Cancellation Decoding of Polar Codes," *IEEE Trans. Commun*., vol. 61, no. 8, Aug. 2013, pp. 3100–07.
[11] A. A. Yazdi and F. R. Kschischang, "A Simplified Successive-Cancellation Decoder for Polar Codes," *IEEE Commun. Lett.*, vol. 15, no. 12, Dec. 2011, pp@. 1378-80.
[12] T. Richardson and R. Urbanke, *Modern Coding Theory*, Cambridge University Press, 2008.
[13] C. Leroux *et al*., "A Semi-Parallel Successive-Cancellation Decoder for Polar Codes," *IEEE Trans. Signal Proc*., vol. 61, no. 2, Jan. 2013, pp. 289–99.
[14] M. Seidl *et al*., "Polar-Coded Modulation," *IEEE Trans. Commun*., vol. 61 , no. 10, Oct. 2013, pp. 4108–19.
[15] K. Chen, K. Niu, and J. Lin, "Improved Successive Cancellation Decoding of Polar Codes," *IEEE Trans. Commun*., vol. 61, no. 8, Aug. 2013, pp. 3100–07.

BIOGRAPHIES

KAI NIU [S'00, M'04] received a B.S. degree in information engineering and a Ph.D. in signal and information processing from Beijing University of Posts and Telecommunications (BUPT), Beijing, China, in 1998 in 2003, respectively. Currently he is an associate professor in the School of Information and Communication Engineering of BUPT. His research interests are in the area of channel coding and broadband wireless communication, particularly on the practical design of polar codes and study of polar decoding algorithms.

KAI CHEN [M'07] received a B.S. degree in information engineering from Beijing University of Posts and Telecommunications (BUPT), Beijing, China, in 2009. He is currently working toward the Ph.D. degree at the School of Information and Communication Engineering of BUPT. His research interests are in information theory, coding theory, and signal processing.

JIARU LIN [M'10] received B.S., M.S. and Ph.D. degrees in signal and information processing from Beijing University of Posts and Telecommunications (BUPT), Beijing, China, in 1982, 1987, and 2001, respectively. From 1991 to 1994 he studied at the Swiss Federal Institute of Technology, Zurich. He is now the professor and Ph.D. supervisor in the School of Information and Communication Engineering, BUPT. His research interests are in wireless communication, cognitive radio network, information theory, and coding.

Q. T. ZHANG [S'84, M'85, SM'95, F'09] received the B.Eng. from Tsinghua University, Beijing, and the M.Eng. from South China University of Technology, Guangzhou, China, both in wireless communications, and the Ph.D. degree in electrical engineering from McMaster University, Hamilton, Ontario, Canada. In 1999 he joined City University of Hong Kong and became a Chair Professor of Information Engineering. His research interest is in wireless communications with a current focus on wireless MIMO, cooperative systems, and cognitive radio. He is an Honorary Professor at Nanjing University of Posts and Telecommunications (NUPT).

The proposed HARQ scheme based on RCP codes can achieve the same performance as the existing schemes based on turbo/LDPC codes by using a simple SC decoding. More flexible HARQ schemes and more powerful decoding algorithms will be the valuable subjects deserved to further explore.