Fourteenth International Workshop on Algebraic and Combinatorial Coding Theory
September 7–13, 2014, Svetlogorsk (Kaliningrad region), Russia *pp. 145–150*

Nested polarized codes: decoding and node selection[1]

Ilya Dumer dumer@ee.ucr.edu
University of California at Riverside, USA

Abstract. We consider decoding for the nested polarized constructions whose end nodes form short Reed-Muller codes or polar codes. Code parameters are then chosen to obtain feasible complexity and to achieve polarization on different end nodes. We also optimize decoding design to achieve higher code rates for moderate lengths.

1 Recursive decoding algorithms

We first describe recursive decoding of codes $R(r,m)$ used on a general memoryless channel $\mathcal{Z}$. Let us map binary symbols $a = 0, 1$ to $(-1)^a$. Then any RM-codeword has the form $\mathbf{c} = (\mathbf{u}, \mathbf{u}\cdot\mathbf{v})$ in $\{1,-1\}^n$, where vector $\mathbf{u}\cdot\mathbf{v}$ is obtained by the component-wise multiplication in $\mathbb{R}$. The received block $\mathbf{x}$ consists of two halves $\mathbf{x}', \mathbf{x}''$ corrupted by noise. Here we first calculate the posterior probability $q_j = \Pr\{c_j = 1 \mid x_j\}$ that 1 is transmitted in position j. To further simplify our calculations, we replace each q_j with an associated quantity $y_j = 2q_j - 1$, which we call the probability *offset*. In particular, it is easy to see that a binary symmetric channel BSC_p with a transition error probability $p = (1-\epsilon)/2$ yields $y(x) = \epsilon x$. In the sequel, we replace the original vector $\mathbf{x}$ with the string $\mathbf{y}$ of n offsets and will split $\mathbf{y}$ into two halves $\mathbf{y}'$ and $\mathbf{y}''$ of length $n/2$. Using $(\mathbf{u}, \mathbf{u}\cdot\mathbf{v})$ construction, the decoder takes the two matching symbols y_i' and y_i'' in two halves of the block for each position $i = 1, ..., n/2$. Our recursive algorithm $\Psi^m_{r,s}(\mathbf{y})$ is outlined below. Step 1 will include all intermediate recursive recalculations, while Step 2 decodes the end nodes.

Note that vector $\mathbf{v}$ is the product of two uncorrupted halves $\mathbf{u}\cdot(\mathbf{uv})$. In Step 1.1, we wish to find the posterior probability $q_i^v = \Pr\{v_i = 1 \mid y_i', y_i''\}$ that $v_i = 1$. Simple recalculations [2] show that the offsets $y_i^v \equiv 2q_i^v - 1$ are derived from the offsets y_i', y_i'' as

$$y_i^v = y_i' y_i''. \tag{1}$$

Let $\mathbf{y}^v$ be a vector of length $n/2$ with symbols y_i^v. We pass $\mathbf{y}^v$ to the next decoding step $\Psi^{m-1}_{r-1,s}$ that gives an output $\hat{\mathbf{v}} \in R(r-1, m-1)$ and its information block $\hat{\mathbf{a}}^v$. In Step 1.2, we assume that Step 1.1 gives the correct vector $\hat{\mathbf{v}}$. Then we have two corrupted versions $\mathbf{x}'$ and $\mathbf{x}''\hat{\mathbf{v}}$ of vector $\mathbf{u}$. Here we find the

[1]This work is supported in part by NSF Grant 1018935

posterior probability $q_i^u = \Pr\{u_i = 1 \mid y'_i, y''_i \hat{v}_i\}$. Here we use symbols $\hat{v}_i$ and find the offsets $y_i^u \equiv 2q_i^u - 1$ as

$$y_i^u = (y'_i + y''_i \hat{v}_i)/(1 + y'_i y''_i \hat{v}_i) \tag{2}$$

We now take vector $\mathbf{y}^u$ with symbols y_i^u and pass it to the next decoding step $\Psi_{r,s}^{m-1}$. The result is some vector $\hat{\mathbf{u}} \in R(r, m-1)$ and its information block $\hat{\mathbf{a}}^u$.

Algorithm $\Psi_{r,s}^m$ for vector $\mathbf{y} = (\mathbf{y}', \mathbf{y}'')$.

1. If $r = s$, go to Step 2. Otherwise:

   1.1. Calculate vector $\mathbf{y}^v$ and
   execute decoding $\hat{\mathbf{v}} = \Psi_{r-1,s}^{m-1}(\mathbf{y}^v)$.
   Pass $\hat{\mathbf{v}}$ and $\hat{\mathbf{a}}^v$ to Step 1.2

   1.2. Calculate vector $\mathbf{y}^u$ and
   execute decoding $\hat{\mathbf{u}} = \Psi_{r,s}^{m-1}(\mathbf{y}^u)$.

   Output decoded components
   $\hat{\mathbf{a}} := (\hat{\mathbf{a}}^v \mid \hat{\mathbf{a}}^u)$; $\hat{\mathbf{c}} := (\hat{\mathbf{u}} \mid \hat{\mathbf{u}}\hat{\mathbf{v}})$.

2. ML-decode $R(s, m)$.
   Return $(\hat{\mathbf{v}}, \hat{\mathbf{a}}^v)$ or $(\hat{\mathbf{u}}, \hat{\mathbf{a}}^u)$.

In this way, algorithm $\Psi_{r,s}^m$ proceeds with recalculations (1) and (2) on the codes of length $n/2$, $n/4$, and so on, until the running parameter r equals s. Then Step 2 performs MLD at the end nodes $R(s, g)$. Here our input $\mathbf{y}$ of length 2^g has the given offsets y_i between posterior probabilities of the two values $c_i = \pm 1$. Thus, each input symbol c_i has posterior probability $\Pr(c_i|y_i) = (1 + c_i y_i)/2$. MLD chooses the codeword $\hat{\mathbf{c}} \in R(s, g)$ with the highest posterior probability

$$P(\mathbf{c} \mid \mathbf{y}) = \prod_{i=1}^{2^g} (1 + c_i y_i)/2 \tag{3}$$

It is easy to verify [2] that the algorithms $\Psi_{m,s}^m$ have low decoding complexity of order $n \log n$ for $s = 0, 1$. Here the case $s = 0$ corresponds to design C and is exactly a "bit-by-bit" cancellation decoder of [1]- [3]. Also, the case $s = 1$ leads to design B that ends at the biorthogonal codes. Note also that recalculations (1) always degrade the two original channels for y'_i, y''_i. By contrast, recalculations (2) combine two noisy copies of presumably the same block and reduce the output BER. Therefore, different paths reproduce channels of different quality. By pruning the noisiest paths we can improve code performance. The breakthrough of [1] shows that the optimal choice of these paths/channels yields the capacity-achieving codes. Our next goal is to try to improve code performance by decoding some nodes $R(s, g)$ instead of individual bits.

2 Design of nested codes

We will now use the description of nested codes from the previous paper. Consider some sets of nested codes $C(m,T)$ of a given code rate $R=k(m,T)/n$. Let $C^s(m,T)$ and $C_g(m,T)$ be two classes of nested codes whose end nodes $R(a,b)$ have restricted parameters $a\le s$ or $b\le g$, respectively. Here $C_0(m,T)$ is the set of classic polar codes. Our decoding $\Psi^m_{r,s}$ can be applied for codes $C^s(m,T)$ and $C_g(m,T)$ with almost no alterations. For example, codes $C_g(m,T)$ have some path-specific end parameters $g(\xi)\le g$ and $s(\xi)$ for each path $\xi\in T$. Therefore, we go from Step 1 to Step 2 if $m=g(\xi)$. Then in Step 2, we use MLD for a path-specific end code $R(\xi)\equiv R(s(\xi),m(\xi))$. Let $\Psi(m,T)$ denote this modified algorithm and let $\Omega(m,T)$ be its complexity. It follows from [2] that $\Omega(m,T)$ has the order of $n\log n$ for codes $C^1(m,T)$. Indeed, here we consider some subset of paths T and apply the same algorithm $\Psi^m_{m,1}$ that has complexity $n\log n$ on the full set of paths $R(m,m)$. Below we slightly extend this statement for codes $C_g(m,T)$. We write $f(m)\prec\varphi(m)$ if $\sup f(m)/\varphi(m)=c$, where $m\to\infty$ and $c<1$. The following simple statement shows that codes $C_g(m,T)$ can also have low complexity at the expense of tight restrictions.

Lemma 1. *Codes $C_g(m,T)$ with $g\prec\log_2 m$ have decoding complexity $\Omega(m,T)\prec n^{1+\ln^{-1}m}$.*

Proof. Recall [2] that all intermediate recalculations (1) and (2) have complexity $n\log_2 n$. Any end node $R(s,g)$ with $g=c\log_2 m$ has length $\eta=2^g=m^c$ and its MLD-complexity is bounded by $2^\eta=o(2^{m/\ln m})$. The number of end nodes, which is bounded by $2^{m-g}<n$ gives the estimate of $\Omega(m,T)$. □

Remark. The above estimate on $\Omega(m,T)$ is essentially tight. Indeed, we will see in the sequel that most good paths pass the nodes $R(s,g)$ of the relatively high orders $s\sim g/2$. In this case, any exponential estimate $2^{c\eta}$ for the MLD complexity will still give a similar estimate on $\Omega(m,T)$. Thus, any essential increase in parameter g is possible only if near-ML decoding can be performed with a much lower complexity.

Lemma 1 shows that codes $C_g(m,T)$ with $g\prec\log_2 m$ yield only a slight complexity overhead over codes $C_0(m,T)$. On the other hand, the following lemma shows that these codes neither degrade nor improve the *asymptotic* bit error rate (BER) for $m\to\infty$. Indeed, consider a channel $\mathcal{Z}$ of capacity C. We say that a sequence of codes $C(m,T)$ of rate $R<C$ is R-polarized on channel $\mathcal{Z}$ if the end nodes on each path $\xi\in T$ yield a vanishing decoding error probability as $m\to\infty$.

Lemma 2. *Any sequence of R-polarized codes $C_0(m,T)$ yields some sequences of ρ-polarized codes $C^1(m,T')$ or τ-polarized codes $C_g(m,T'')$ of rates $\rho\to\tau\to R$ for $m\to\infty$ and $g\le\log_2 m$.*

Proof. We first convert a polar code $C_0(m,T)$ to $C^1(m,T')$. There are $R2^m$ paths $\xi\in T$. Most paths ξ have weight w such that $|w-m/2|\le\sqrt{m}$

$\log_2 m$ as $m \to \infty$. In turn, most of the latter paths have weight $w-1$ on some length $\ell(\xi) > m - 3\sqrt{m}\,\log_2 m$. Then these paths end with suffixes of weight 1 on the last $g(\xi) \le 3\sqrt{m}\,\log_2 m$ positions. We will now end all remaining paths ξ on the length $\ell(\xi)$ and use the end codes $R(1, g(\xi))$. This gives a code $C^1(m, T')$ of some rate $R - o_1(1)$. Next, note that paths $\xi \in T'$ have the length $\ell(\xi) \sim m$, similar to the original paths. Then these paths of length $\ell(\xi)$, except a fraction $o_2(1)$ of them, achieve polarization according to [1]. Thus, some fraction $\rho = R - o_1(1) - o_2(1)$ of paths $\xi \in T$ can be decoded by codes $R(1, g(\xi))$ with a vanishing error probability.

We use similar arguments for codes $C_g(m, T'')$ with $g \le \log_2 m$. Again, we begin with the set T. Consider any path $\xi \in T$ of length m at the near-full length $\ell = m - \lceil \log_2 m \rceil$. All paths in T except a fraction $o_3(1)$ of them, achieve polarization at the length ℓ with a vanishing error probability p. We can also assume that $p = o(1/m)$. The set T'' of these paths will now be terminated with some end nodes (s, g), where $s \le g$ and $g \le \log_2 m$. Then ML decoding of a code $R(s, g)$ gives a vanishing error probability for $m \to \infty$. □

The above design discards all $n(1-R)$ high–noise paths ξ that have error probability $p(\xi) \to 1/2$ as $m \to \infty$. By contrast, $p(\xi)$ can differ substantially from $1/2$ on some noisy paths of small length m. Then these paths can be efficiently decoded using some low-rate codes $R(s, g)$ of fixed order s. We will now estimate BER of some noisy paths and their end nodes $R(s, g)$ in order to add some of them to the conventional polar design. Here we can use an efficient algorithm of [4] that tightly evaluates the channel error probability $p\,(\xi)$ obtained on an arbitrary path ξ. Given a path $\xi = (\xi_1, ..., \xi_\ell)$ with some end node $R(s, g)$, let $\overline{\xi} = (\xi, \mathbf{1}^{g-s})$ be a path of length $m - s$ that extends ξ with $g - s$ ones, and $\xi^* = (\xi, \mathbf{1}^g)$ be a full path that extends ξ with g ones.

Lemma 3. *Consider a path ξ that ends at some node $R(s, g)$ of size $M(\xi)$ and the corresponding code $C_g^s(\xi) = \{\mathbf{c}(\xi, \mathbf{c}_\ell) \mid \mathbf{c}_\ell \in R(s, g)\}$. Then ML-decoding of this code has error probability $P(\xi)$ bounded as*

$$P(\xi) \le (M(\xi) - 1)p\left(\overline{\xi}\right) \tag{4}$$

$$P(\xi) \le (M(\xi)/2 - 1)(2p\left(\overline{\xi}\right) - p\,(\xi^*)) + p\,(\xi^*) \tag{5}$$

Proof. Any two codewords $\mathbf{c}_\ell, \mathbf{b}_\ell \in R(s, g)$ differ on some subset I_w of $w \ge d$ positions, where $d = 2^{g-s}$. A decoding error $\mathbf{c}_\ell \mapsto \mathbf{b}_\ell$ can be regarded as an error in a repetition $[w, 1, w]$-code considered on the set I_w. Also, the error probability of this code is upper bounded by the error probability of a $[d, 1, d]$-code. Thus, we can estimate the error probability $P\{\mathbf{c}_\ell \mapsto \mathbf{b}_\ell\}$ by an error probability of a path ξ that ends at a repetition code $R(0, g - s)$. This code is defined by a path $\mathbf{1}^{g-s}$ in our encoding. Thus, path ξ with an end node $R(0, g - s)$ represents the extended path $\overline{\xi}$. Then $P\{\mathbf{c}_\ell \mapsto \mathbf{b}_\ell\} \le p\left(\overline{\xi}\right)$ and we obtain estimate (4). To obtain (5), decompose code $R(s, g)$ in $M(\xi)/2$ couples

$\mathbf{b}_\ell$, $\mathbf{b}'_\ell$ of opposite codewords, including a couple $\mathbf{c}_\ell, \mathbf{c}'_\ell$. Then the event $\{\mathbf{c}_\ell \mapsto \mathbf{c}'_\ell\}$ has probability $p\,(\xi^*)$. The compound error event $\{\mathbf{c}_\ell \mapsto \mathbf{b}_\ell\} \cup \{\mathbf{c}_\ell \mapsto \mathbf{b}'_\ell\}$ has probability bounded by $2p\left(\overline{\xi}\right) - p\,(\xi^*)$. This gives the union bound in (5). □

Note that (5) is essentially tight for the shortest codes, such as $R(0,0)$ and $R(1,1)$, which will form the bulk of our design. Bounds (4) and (5) are easily generalized for multi-path codes $C(m,T)$. Here we choose different parameters $s(\xi)$, $g(\xi)$ and $M(\xi)$ for different paths $\xi \in T$ and obtain different paths $\overline{\xi} = (\xi, \mathbf{1}^{\mathbf{g}(\xi)-\mathbf{s}(\xi)})$ and $\xi^* = (\xi, \mathbf{1}^g)$. This gives a similar estimate for the error probability $P(m,T)$ of code $C(m,T)$,

$$P(m,T) \le \sum_{\xi \in T} P(\xi) \tag{6}$$

Discussion. Any node $R(s,g)$ in code $C(m,T)$ in essence forms a collection of $M(\xi)$ paths (ξ,η) of length m that have the common prefix ξ of length $m-g$ but different suffixes η of weight $wt(\eta) \ge g-s$. ML-decoding essentially replaces different error probabilities on the paths (ξ,η) with the same probability $p\left(\overline{\xi}\right)$ for the single path $(\xi, \mathbf{1}^{g-s})$. The premise of our design rests on the fact that the suffix $\mathbf{1}^{g-s}$ combines $g-s$ good channels and therefore can better many (but not all) suffixes η, particularly those that have zeros in the beginning. To further this design, we can also refine loose upper bound (5), (6) using the weight spectra M_w of our codes $R(s,g)$. Then the single suffix $\mathbf{1}^{g-s}$ used in $p\left(\overline{\xi}\right)$ can be replaced with various suffixes ξ_w that correspond to different codes $[w,1,w]$. The overall estimate (5) is then replaced with a tighter bound

$$P(\xi) \le \sum_{d \le w < n} (M_w(\xi)/2 - 1)(2p\,(\xi, \xi_w) - p\,(\xi^*)) + p\,(\xi^*) \tag{7}$$

Optimization algorithm. Consider a general code $C(m,T)$ as a collection of different paths ξ and their end codes $R(s,g)$. Let $b(\xi) = P(\xi)/k(\xi)$ denote an output BER obtained for MLD of code $R(s,g)$. Let each $b(\xi)$ satisfy some threshold restriction $b(\xi) \le \theta$, which in turn restricts BER for the overall code $C(m,T)$. Given a channel $\mathcal{Z}$, we wish to consider various paths ξ and assign the end nodes $R(s,g)$ that can satisfy the threshold θ under MLD.

To optimize code $C(m,T)$, we perform a greedy, tree-like search over different paths ξ. This search includes an outer cycle over an increasing order $s = 0, ..., t$ of code $R(s,g)$. Here $t \le \lceil \log_2 m \rceil$ is a parameter. For each step s, we also employ the inner cycle over the increasing length $\ell = 1, ..., m-s$. Length ℓ also defines parameter $g = m - \ell$ of an end code $R(s,g)$.

The procedure begins with an empty list of paths T at the first outer step $s = 0$. Given some inner step ℓ, we seek any path ξ of length ℓ that gives BER $b(\xi) \le \theta$ at the single-bit repetition node $R(0,g)$ that corresponds to the suffix $\eta = 1^g$. Note that path ξ and its suffix form a single path of length m. Thus,

our step $s = 0$ runs over all individual paths ξ of length m and selects all paths $\xi \in T$ of the classic polar design. Since paths ξ are collected at different lengths ℓ, we count each path at its first appearance. Then any path ξ is recorded in the list T with its end parameters $(0, g)$.

We use the same procedure in other steps. For each outer step s and inner step $\ell \le m - s$, we take $g = m - \ell$ and seek paths ξ that give the required BER for MLD of codes $R(s, g)$. For $s \ge 1$, any such path ξ yields a subset of extended paths (ξ, η) . Here we use $k(\xi)$ different suffixes η that correspond to the end node $R(s, g)$. Next, we compare the extended paths (ξ, η) with the similar extensions (ξ', η') of the previously recorded paths $\xi' \in T$. Then path ξ and its end parameters (s, g) are added to the list T if ξ generates at least one new extension (ξ, η) . Thus, we expand the classic polar code if some new paths of length m emerge in MLD of non-repetition codes $R(s, g)$.

Remark. Note that a new path ξ is decoded by some RM code $R(s, g)$ even if ξ includes some extensions (ξ', η') that were already decoded by the previous paths. Then the new extensions (ξ, η) of a path ξ form a smaller polar code within the RM code $R(s, g)$. In this case, we use RM codes for MLD but can take a smaller size of a polar code to tighten BER bounds (4)-(7).

Open problems. Design of nested constructions can be advanced as follows. Firstly, we need to construct feasible near-MLD algorithms that can extend Lemma 1 for relatively large codes $R(s, g)$ with $g > \log_2 m$. Secondly, we also need to tighten bounds (4)-(7) to achieve any material improvement over classic polar design. Indeed, estimates (4)-(7) use a union bound taken over all codewords $\mathbf{b}_\ell$. For the end nodes $R(s, g)$ with small distance 2^{g-s} these bounds yield a multiple overestimate of the actual error probabilities. Further improvements can also use list decoding for nested codes $C(m, T)$, along the lines of the algorithms used in [2, 3]. Finally, this design can use other end nodes different from RM and polar codes.

References

[1] E. Arikan, "Channel polarization: A method for constructing capacity-achieving codes for symmetric binary-input memoryless channels," IEEE Trans. Info. Theory, 55 , pp. 3051-3073, July 2009.

[2] I. Dumer and K. Shabunov, "Recursive list decoding of Reed-Muller codes", Information, Coding and Mathematics, ed. M. Blaum, P. Farrell, and H.C.A. van Tilborg, Kluwer, Boston, 2002, pp. 279-298.

[3] I. Dumer and K. Shabunov, "Soft decision decoding of Reed-Muller codes: recursive lists," IEEE Trans. Info. Theory, 52, pp. 1260-1266, March 2006.

[4] I. Tal and A. Vardy, "How to construct polar codes," arXiv:1105.6164v1, 2011