A score function for sequential decoding of polar codes

Peter Trifonov
Saint Petersburg Polytechnic University, Russia
Email: petert@dcn.icc.spbstu.ru

***Abstract*—A novel score function is proposed for sequential decoding of polar codes. Significant reduction of the average decoding complexity is achieved by biasing the path metrics in the min-sum version of the stack successive cancellation decoding algorithm with its expected value. The proposed approach can be also used for near-ML decoding of short extended BCH codes.**

I. Introduction

Polar codes were recently shown to be able to achieve the symmetric capacity of memoryless channels, while having low-complexity construction, encoding and decoding algorithms [1]. List decoding enables near maximum likelihood decoding of polar codes with complexity $O(Ln\log n)$, where n is code length and L is list size [2]. Polar codes concatenated with a CRC outer code [2] and polar subcodes [3] under list decoding were shown to outperform LDPC and turbo codes.

However, the complexity of the Tal-Vardy list decoding algorithm turns out to be rather high. It can be reduced by employing stack decoding [4], [5]. These methods avoid construction of many useless low-probability paths in the code tree. In this paper we show that careful weighting of paths of different length enables significant reduction the computational complexity of the decoder. The proposed path score function aims to estimate the conditional probability of the most likely codeword of a polar code, which may be obtained as a continuation of the considered path in the code tree. We show that such function can be well approximated by the path score of the min-sum list SC decoder biased by its expected value.

II. Background

A. Polar codes

$(n=2^m,k)$ polar code over $\mathbb{F}_2$ is a linear block code generated by k rows of matrix $A_m=B_mF^{\otimes m}$, where B_m is the bit reversal permutation matrix, $F=\begin{pmatrix}1&0\\1&1\end{pmatrix}$, and $\otimes m$ denotes m-times Kronecker product of the matrix with itself [1]. Hence, a codeword of a classical polar code is obtained as $c_0^{n-1}=u_0^{n-1}A_m$, where $a_s^t=(a_s,\dots,a_t)$, $u_i=0$ for $i\in\mathcal{F}$, $\mathcal{F}\subset\{0,\dots,n-1\}$ is the set of $n-k$ frozen symbol indices, and the remaining symbols of the information vector u_0^{n-1} are set to the data symbols being encoded.

Let $U_0,\dots,U_{n-1}$ and $Y_0,\dots,Y_{n-1}$ be the random variables corresponding to the input symbols of the polarizing transformation A_m, and output symbols of a memoryless output-symmetric channel, respectively. It is possible to show that matrix A_m transforms the original binary input memoryless symmetric channel $W_0^{(0)}(Y|C)=W(Y|C)$ into bit subchannels $W_m^{(i)}(\mathbf{Y},U_0^{i-1}|U_i)$, the capacities of these subchannels converge with m to 0 or 1, and the fraction of subchannels with capacity close to 1 converges to the capacity of $W_0^{(0)}(Y|C)$. Here $\mathbf{Y}=Y_0^{n-1}$, and $C\in\mathbb{F}_2$ is the random variable corresponding to channel input.

The typical approach to construction of an (n,k) polar code assumes that $\mathcal{F}$ is the set of $n-k$ indices i of bit subchannels $W_m^{(i)}(Y_0^{n-1},U_0^{i-1}|U_i)$ with the highest error probability. It was suggested in [3] to set some frozen symbols $U_i, i\in\mathcal{F}$, not to zero, but to linear combinations of other symbols. This results in codes with improved minimum distance, and the obtained codes are referred to as polar subcodes.

B. The successive cancellation decoding algorithm

Decoding of polar (sub)codes can be implemented by the successive cancellation (SC) algorithm. It is convenient to describe the SC algorithm in terms of probabilities $W_m^{(i)}\left\{U_0^i=v_0^i|\mathbf{Y}=y_0^{n-1}\right\}=W_m^{(i)}\left\{v_0^i|y_0^{n-1}\right\}$ of transmission of various vectors $v_0^{n-1}A_m$ with given values v_0^i, provided that the receiver observes a noisy vector y_0^{n-1}, i.e.

$$W_m^{(i)}\left\{v_0^i|y_0^{n-1}\right\}=\frac{W_m^{(i)}(y_0^{n-1},v_0^{i-1}|v_i)}{2W(y_0^{n-1})}$$
$$=\sum_{v_{i+1}^{n-1}}W_m^{(n-1)}\left\{v_0^{n-1}|y_0^{n-1}\right\}=\sum_{v_{i+1}^{n-1}}\prod_{j=0}^{n-1}W\left\{(v_0^{n-1}A_m)_j|y_j\right\} \tag{1}$$

At phase i the SC decoder makes decision

$$\widehat{u}_i=\begin{cases}\arg\max_{v_i\in\mathbb{F}_2}W_m^{(i)}\left\{\widehat{u}_0^{i-1}.v_i|y_0^{n-1}\right\}, & i\notin\mathcal{F}\\ \text{the frozen value of } u_i, & \text{otherwise,}\end{cases}$$

where $a.b$ denotes a vector obtained by appending b to a.

C. Improved decoding algorithms

The SC algorithm does not provide maximum likelihood decoding. A successive cancellation list (SCL) decoding algorithm was suggested in [2], and shown to achieve substantially better performance with complexity $O(Ln\log n)$. large values of L are needed to implement near-ML decoding of polar subcodes and polar codes with CRC. This makes practical implementations of such list decoders very challenging.

In practice one does not need to obtain a list of codewords, but just a single most probable one. The Tal-Vardy algorithm for polar codes with CRC examines the elements in the obtained list, and discards those with invalid checksums. This algorithm can be easily tailored to process the dynamic freezing constraints used in the construction of polar subcodes [3] before the decoder reaches the last phase, so that the output list contains only valid codewords. However, even in this case $L-1$ codewords are discarded from the obtained list, so most of the work performed by the Tal-Vardy decoder is just wasted.

This problem was addressed in [4], where a generalization of the stack algorithm to the case of polar codes was suggested. It provides lower average decoding complexity compared to the Tal-Vardy algorithm. In this paper we revise the stack decoding algorithm for polar (sub)codes, and show that its complexity can be substantially reduced.

III. Path score

A. Stack decoding algorithm

Let u_0^{n-1} be the input vector used by the transmitter. Given a received noisy vector y_0^{n-1}, the proposed decoding algorithm constructs sequentially a number of partial candidate information vectors $v_0^{\phi-1} \in \mathbb{F}_2^{\phi}, \phi \leq n$, evaluates how close their continuations v_0^{n-1} may be to the received sequence, and eventually produces a single codeword, being a solution of the decoding problem.

The stack decoding algorithm [6], [7], [4], [5] employs a priority queue[1] (PQ) to store paths together with their scores. A PQ is a data structure, which contains tuples $(M, v_0^{\phi-1})$, where $M = M(v_0^{\phi-1}, y_0^{n-1})$ is the score of path $v_0^{\phi-1}$, and provides efficient algorithms for the following operations [8]:

- push a tuple into the PQ;
- pop a tuple $(M, v_0^{\phi-1})$ with the highest M;
- remove a given tuple from the PQ.

We assume here that the PQ may contain at most D elements.

In the context of polar codes, the stack decoding algorithm operates as follows:

1) Push into the PQ the root of the tree with score 0. Let $t_0^{n-1} = 0$.
2) Extract from the PQ a path $v_0^{\phi-1}$ with the highest score. Let $t_\phi \leftarrow t_\phi + 1$.
3) If $\phi = n$, return codeword $v_0^{n-1} A_m$ and terminate.
4) If the number of valid (i.e. those satisfying freezing constraints) children v_0^{ϕ} of path $v_0^{\phi-1}$ exceeds the amount of free space in the PQ, remove from it the element with the smallest score.
5) Compute the scores $M(v_0^{\phi}, y_0^{n-1})$ of valid children v_0^{ϕ} of the extracted path, and push them into the PQ.
6) If $t_\phi \geq L$, remove from PQ all paths $v_0^{j-1}, j \leq \phi$.
7) Go to step 2.

In what follows, one iteration means one pass of the above algorithm over steps 2–7. Variables t_ϕ are used to ensure that the worst-case complexity of the algorithm does not exceed that of a list SC decoder with list size L.

[1] A PQ is commonly called "stack" in the sequential decoding literature. However, the implementation of the considered algorithm relies on Tal-Vardy data structures [2], which make use of the true stacks. Therefore, we employ the standard terminology of computer science.

The parameter L has the same impact on the performance of the decoding algorithm as the list size in the Tal-Vardy algorithm, since it imposes an upper bound on number of paths t_ϕ considered by the decoder at each phase ϕ. Step 6 ensures that the algorithm terminates in at most Ln iterations. This is also an upper bound on the number of entries stored in the PQ. However, the algorithm can work with PQ of much smaller size D. Step 4 ensures that this size is never exceeded.

B. Score function

There are many possible ways to define a score function for sequential decoding. In general, this should be done so that one can perform meaningful comparison of paths $v_0^{\phi-1}$ of different length ϕ. The classical Fano metric for sequential decoding of convolutional codes is given by the probability

$$P\left\{\mathcal{M}|y_0^{n-1}\right\} = \frac{P\left\{\mathcal{M}, y_0^{n-1}\right\}}{\prod_{i=0}^{n-1} W(y_i)},$$

where $\mathcal{M}$ is a variable-length message (i.e. a path in the code tree), and $W(y_i)$ is the probability measure induced on the channel output alphabet when the channel inputs follow some prescribed (e.g. uniform) distribution [9]. In the context of polar codes, a straightforward implementation of this approach would correspond to score function

$$M_1(v_0^{\phi-1}, y_0^{n-1}) = \log W_m^{(\phi-1)}\left\{v_0^{\phi-1}|y_0^{n-1}\right\}.$$

This is exactly the score function used in [4]. However, there are several shortcomings in such definition:

1) Although the value of the score does depend on all $y_i, 0 \leq i < n$, it does not take into account freezing constraints on symbols $u_i, i \in \mathcal{F}, i \geq \phi$. As a result, there may exist incorrect paths $v_0^{\phi-1} \neq u_0^{\phi-1}$, which have many low-probability continuations $v_0^{n-1}, v_\phi^{n-1} \in \mathbb{F}_2^{n-\phi}$, such that the probability
$$W_m^{(\phi-1)}\left\{v_0^{\phi-1}|y_0^{n-1}\right\} = \sum_{v_\phi^{n-1}} W_m^{(n-1)}\left\{v_0^{n-1}|y_0^{n-1}\right\}$$
becomes high, and the stack decoder is forced to expand such a path. This is not a problem for convolutional codes, where the decoder may recover after an error burst, i.e. obtain a codeword identical to the transmitted one, except for a few closely located symbols.
2) Due to freezing constraints, not all vectors $v_0^{\phi-1}$ correspond to valid paths in the code tree. This does not allow one to fairly compare the probabilities of paths of different lengths with different number of frozen symbols.
3) Computing probabilities $W_m^{(\phi-1)}\left\{v_0^{\phi-1}|y_0^{n-1}\right\}$ involves expensive multiplications and is prone to numeric errors.

The first of the above problems can be addressed by considering only the most probable continuation of path $v_0^{\phi-1}$, i.e. the score function can be defined as

$$M_2(v_0^{\phi-1}, y_0^{n-1}) = \max_{v_\phi^{n-1} \in \mathbb{F}_2^{n-\phi}} \log W_m^{(n-1)}\left\{v_0^{n-1}|y_0^{n-1}\right\}.$$

Observe that maximization is performed over last $n-\phi$ elements of vector v_0^{n-1}, while the remaining ones are given by $v_0^{\phi-1}$. Let us further define

$$\mathbf{V}(v_0^{\phi-1},y_0^{n-1})=\arg\max_{\substack{w_0^{n-1}\in\mathbb{F}_2^n\\ w_0^{\phi-1}=v_0^{\phi-1}}}\log W_m^{(n-1)}\left\{w_0^{n-1}|y_0^{n-1}\right\},$$

i.e. $M_2(v_0^{\phi-1},y_0^{n-1})=\log W_m^{(n-1)}\left\{\mathbf{V}(v_0^{\phi-1},y_0^{n-1})|y_0^{n-1}\right\}$.

As shown below, employing such score function already provides significant reduction of the average number of iterations at the expense of a negligible performance degradation. Furthermore, it turns out that this score is exactly equal to the one used in the min-sum version of the Tal-Vardy list decoding algorithm [10], i.e. it can be computed in a very simple way.

To address the second problem, we need to evaluate the probabilities of vectors v_0^{n-1} under freezing conditions. To do this, consider the set $C(\phi)$ of valid length-ϕ prefixes of the input vectors of the polarizing transformation, i.e. vectors $v_0^{\phi-1}$ satisfying the corresponding freezing constraints. Let us further define the set of their most likely continuations, i.e.

$$\overline{C}(\phi)=\left\{\mathbf{V}(v_0^{\phi-1},y_0^{n-1})|v_0^{\phi-1}\in C(\phi)\right\}.$$

For any $v_0^{n-1}\in\overline{C}(\phi)$ the probability of transmission of $v_0^{n-1}A_m$, under condition of $v_0^{\phi-1}\in C(\phi)$ and given the received vector y_0^{n-1}, equals

$$\mathbb{W}\left\{v_0^{n-1}|y_0^{n-1},C(\phi)\right\}=\frac{W_m^{(n-1)}\left\{U_0^{n-1}=v_0^{n-1}|y_0^{n-1}\right\}}{W_m^{(n-1)}\left\{U_0^{n-1}\in\overline{C}(\phi)|y_0^{n-1}\right\}}.$$

Hence, an ideal score function could be defined as

$$\mathbb{M}(v_0^{\phi-1},y_0^{n-1})=\log\mathbb{W}\left\{\mathbf{V}(v_0^{\phi-1},y_0^{n-1})|y_0^{n-1},C(\phi)\right\}.$$

Observe that this function is defined only for vectors $v_0^{\phi-1}\in C(\phi)$, i.e. those satisfying freezing constraints up to phase ϕ.

Unfortunately, there is no simple and obvious way to compute $\pi(\phi,y_0^{n-1})=W_m^{(n-1)}\left\{U_0^{n-1}\in\overline{C}(\phi)|y_0^{n-1}\right\}$. Therefore, we have to develop an approximation. It can be seen that

$$\begin{aligned}\pi(\phi,y_0^{n-1})=&W_m^{(n-1)}\left\{\mathbf{V}(u_0^{\phi-1})|y_0^{n-1}\right\}+\\ &\underbrace{\sum_{\substack{v_0^{\phi-1}\in C(\phi)\\ v_0^{\phi-1}\neq u_0^{\phi-1}}}W_m^{(n-1)}\left\{\mathbf{V}(v_0^{\phi-1})|y_0^{n-1}\right\}}_{\mu(u_0^{\phi-1},y_0^{n-1})}.\end{aligned}\qquad(2)$$

Observe that $p=\mathbf{E}_{\mathbf{Y}}\left[\frac{\mu(u_0^{n-1},\mathbf{Y})}{\pi(\phi,\mathbf{Y})}\right]$ is the total probability of incorrect paths at phase ϕ of the min-sum version of the Tal-Vardy list decoding algorithm with infinite list size, under the condition of all processed freezing constraints being satisfied. We consider decoding of polar (sub)codes, which are constructed to have low list SC decoding error probability even for small list size in the considered channel $W(y|c)$. Hence, it can be assumed that $p\ll 1$. This implies that with high probability $\mu(u_0^{\phi-1},y_0^{n-1})\ll W_m^{(n-1)}\left\{\mathbf{V}(u_0^{\phi-1})|y_0^{n-1}\right\}$, i.e. $\pi(\phi,y_0^{n-1})\approx W_m^{(n-1)}\left\{U_0^{n-1}=\mathbf{V}(u_0^{\phi-1})|y_0^{n-1}\right\}$.

However, a real decoder cannot compute this value, since the transmitted vector u_0^{n-1} is not available at the receiver side. Therefore, we propose to further approximate the logarithm of the first term in (2) with its expected value over $\mathbf{Y}$, i.e.

$$\log\pi(\phi,y_0^{n-1})\approx\Psi(\phi)=\mathbf{E}_{\mathbf{Y}}\left[\log W_m^{(\phi-1)}\left\{\mathbf{V}(u_0^{\phi-1})|\mathbf{Y}\right\}\right]$$

Observe that this value depends only on ϕ and underlying channel $W(Y|C)$, and can be pre-computed offline.

Hence, instead of the ideal score function $\mathbb{M}(v_0^{\phi-1},y_0^{n-1})$, we propose to use an approximate one

$$M_3(v_0^{\phi-1},y_0^{n-1})=M_2(v_0^{\phi-1},y_0^{n-1})-\Psi(\phi).\qquad(3)$$

Observe that for the correct path $v_0^{\phi-1}=u_0^{\phi-1}$ one has $\mathbf{E}_{\mathbf{Y}}\left[M_3(v_0^{\phi-1},\mathbf{Y})\right]=0$.

C. Computing the score function

Consider computing

$$R_m^{(\phi-1)}(v_0^{\phi-1},y_0^{n-1})=M_2(v_0^{\phi-1},y_0^{n-1}).$$

Let the modified log-likelihood ratios be defined as

$$S_m^{(\phi)}(v_0^{\phi-1}|y_0^{n-1})=R_m^{(\phi)}(v_0^{\phi-1}.0,y_0^{n-1})-R_m^{(\phi)}(v_0^{\phi-1}.1,y_0^{n-1}).\qquad(4)$$

It can be seen that

$$\begin{aligned}R_m^{(\phi)}(v_0^{\phi},y_0^{n-1})=&R_m^{(\phi-1)}(v_0^{\phi-1},y_0^{n-1})\\ &+\tau(S_m^{(\phi)}(v_0^{\phi-1}|y_0^{n-1}),v_\phi),\end{aligned}\qquad(5)$$

where

$$\tau(S,v)=\begin{cases}0, & \text{if } \operatorname{sgn}(S)=(-1)^v\\ -|S|, & \text{otherwise.}\end{cases}$$

is the penalty function. Indeed, let $\tilde{v}_0^{n-1}=\mathbf{V}(v_0^{\phi-1})$. If $v_\phi=\tilde{v}_\phi$, then the most probable continuations of $v_0^{\phi-1}$ and v_0^{ϕ} are identical. Otherwise, $-\left|S_m^{(\phi)}(v_0^{\phi-1}|y_0^{n-1})\right|$ is exactly the difference between the log-probability of the most likely continuations of $v_0^{\phi-1}$ and v_0^{ϕ}.

The initial value for recursion (5) is given by

$$R_m^{(-1)}(y_0^{n-1})=\log\prod_{i=0}^{n-1}W\left\{C=\hat{c}_i|Y=y_i\right\},$$

where $\hat{c}_i$ is the hard decision corresponding to y_i. However, this value can be replaced with 0, since it does not affect the selection of paths in the stack algorithm.

Since $R_m^{(\phi-1)}(v_0^{\phi-1},y_0^{n-1})$ is obtained by maximization of $W_m^{(n-1)}\left\{v_0^{n-1}|y_0^{n-1}\right\}$ over v_ϕ^{n-1}, it can be seen that

$$\begin{aligned}&R_\lambda^{(2i)}(v_0^{2i},y_0^{N-1})=\\ &\max_{v_{2i+1}}\left(R_{\lambda-1}^{(i)}\left(v_{0,e}^{2i+1}\oplus v_{0,o}^{2i+1},y_0^{\frac{N}{2}-1}\right)+R_{\lambda-1}^{(i)}\left(v_{0,o}^{2i+1},y_{\frac{N}{2}}^{N-1}\right)\right),\\ &R_\lambda^{(2i+1)}(v_0^{2i+1}|y_0^{N-1})=\\ &\qquad R_{\lambda-1}^{(i)}\left(v_{0,e}^{2i+1}\oplus v_{0,o}^{2i+1},y_0^{\frac{N}{2}-1}\right)+R_{\lambda-1}^{(i)}\left(v_{0,o}^{2i+1},y_{\frac{N}{2}}^{N-1}\right),\end{aligned}$$

where $N=2^\lambda$, $0<\lambda\leq m$, and initial values for these recursive expressions are given by $R_0^{(0)}(b,y_j)=\log W_0^{(0)}\{b|y_j\}$, $b\in\{0,1\}$. From (4) one obtains

$$\begin{aligned}S_\lambda^{(2i)}(v_0^{2i-1}|y_0^{2^\lambda-1}) =&\max(J(0)+K(0),J(1)+K(1))-\\&\max(J(1)+K(0),J(0)+K(1))\\=&\max(J(0)-J(1)+K(0)-K(1),0)-\\&\max(K(0)-K(1),J(0)-J(1))\\S_\lambda^{(2i+1)}(v_0^{2i}|y_0^{2^\lambda-1}) =&J(v_{2i})+K(0)-J(v_{2i}+1)-K(1)\end{aligned}$$

where $J(c)=R_{\lambda-1}^{(i)}((v_{0,e}^{2i-1}\oplus v_{0,o}^{2i-1}).c|y_0^{2^{\lambda-1}-1})$, $K(c)=R_{\lambda-1}^{(i)}(v_{0,o}^{2i-1}.c|y_{2^{\lambda-1}}^{2^\lambda-1})$. Observe that

$$J(0)-J(1)=a=S_{\lambda-1}^{(i)}(v_{0,e}^{2i-1}\oplus v_{0,o}^{2i-1}|y_0^{2^{\lambda-1}-1})$$

and

$$K(0)-K(1)=b=S_{\lambda-1}^{(i)}(v_{0,o}^{2i-1}|y_{2^{\lambda-1}}^{2^\lambda-1})$$

It can be obtained from these expressions that the modified log-likelihood ratios are given by

$$\begin{aligned}S_\lambda^{(2i)}(v_0^{2i-1}|y_0^{2^\lambda-1}) =&Q(a,b)=\operatorname{sgn}(a)\operatorname{sgn}(b)\min(|a|,|b|),\\S_\lambda^{(2i+1)}(v_0^{2i}|y_0^{2^\lambda-1}) =&P(v_{2i},a,b)=(-1)^{v_{2i}}a+b.\end{aligned}$$

The initial values for this recursion are given by $S_0^{(0)}(y_i)=\log\frac{W\{0|y_i\}}{W\{1|y_i\}}$. These expressions can be readily recognized as the min-sum approximation of the list SC algorithm[10]. However, these are also the exact values, which reflect the probability of the most likely continuation of a given path $v_0^{\phi-1}$ in the code tree.

D. The bias function

The function $\Psi(\phi)$ is equal to the expected value of the logarithm of the probability of a length-ϕ part of the correct path, i.e. the path corresponding to the vector u_0^{n-1} used by the encoder. Employing this function enables one to estimate how far a particular path $v_0^{\phi-1}$ has diverted from the expected behaviour of a correct path. The bias function can be computed offline under the assumption of zero codeword transmission. Indeed, in this case the cumulative density functions $F_\lambda^{(i)}(x)$ of $S_\lambda^{(i)}$ are given by [11]

$$F_\lambda^{(2i)}(x)=\begin{cases}2F_{\lambda-1}^{(i)}(x)(1-F_{\lambda-1}^{(i)}(-x)), & x<0\\ 2F_{\lambda-1}^{(i)}(x)-(F_{\lambda-1}^{(i)}(-x))^2-(F_{\lambda-1}^{(i)}(x))^2, & x\geq 0\end{cases}$$

$$F_\lambda^{(2i+1)}(x)=\int_{-\infty}^{\infty}F_{\lambda-1}^{(i)}(x-y)dF_{\lambda-1}^{(i)}(y),$$

where $F_0^{(0)}(x)$ is the CDF of the channel output LLRs. Hence,

$$\Psi(\phi)=-\sum_{i=0}^{\phi-1}\int_{-\infty}^{0}F_m^{(i)}(x)dx. \quad (6)$$

The bias function $\Psi(\phi)$ depends only on m and channel properties, so it can be used for decoding of any polar (sub)code of a given length.

TABLE I
AVERAGE DECODING COMPLEXITY OF $(1024,512,28)$ CODE WITH $L=32$, $\times10^3$ OPERATIONS

E_b/N_0, dB	Summations		Comparisons	
	Proposed	[5]	Proposed	[5]
0.5	63.2	133	122.5	218
1	34.8	73	55.6	122
1.5	16	32	21.9	54
2	8.8	18	12.0	31

E. Complexity analysis

The algorithm presented in Section III-A extracts from the PQ length-ϕ paths at most L times. At each iteration it needs to calculate the LLR $S_m^{(\phi)}(v_0^{\phi-1}|y_0^{n-1})$. Intermediate values for these calculations can be reused in the same way as in [2]. Hence, LLR calculations require at most $O(Ln\log n)$ operations. However, simulation results presented below suggest that the average complexity of the proposed algorithm is substantially lower, and at high SNR approaches $O(n\log n)$, the complexity of the SC algorithm.

IV. NUMERIC RESULTS

The performance and complexity of the proposed decoding algorithm were investigated in the case of BPSK modulation, AWGN channel and polar subcode (PS) [3]. The size of the priority queue was set in all cases to $D=Ln$.

Figure 1 illustrates the decoding error probability and average number of iterations performed by the sequential decoder for the case of the path scores M_1, M_2 and M_3. The first two scores correspond to the Niu-Chen stack decoding algorithm and its min-sum version [4], which, for sufficiently large size D of the priority queue, achieves the same performance as the Tal-Vardy algorithm with the same value of L.

Observe that employing score M_2 results in a marginal performance loss, but significant reduction of the average number of iterations performed by the decoder. This is due to existence of multiple paths v_0^{n-1} with low probability $W_m^{(n-1)}\left\{v_0^{n-1}|y_0^{n-1}\right\}$, which add up (see (1)) to non-negligible probabilities $W_m^{(\phi-1)}\left\{v_0^{\phi-1}|y_0^{n-1}\right\}$. Employing path score M_1 causes the decoder to inspect many incorrect paths $v_0^{\phi-1}$. At sufficiently high SNR the most probable continuation $\mathbf{V}(v_0^{\phi-1})$ of a path extracted at some phase from the PQ with high probability satisfies all freezing constraints, so that the value given by M_2 score function is close to the final path score. This enables the decoder to avoid visiting many incorrect paths in the code tree. However, if the most likely continuation of a path does not satisfy some freezing constraints, then the decoder may start exploring an incorrect subtree and eventually kill the correct path. Such events are responsible for the performance loss with respect to M_1 score.

Even more significant complexity reduction is obtained by employing the proposed path score M_3. It enables one to properly compare the probabilities of paths $v_0^{\phi-1}$ of different length ϕ. This results in an order of magnitude reduction of the average number of iterations. Observe that the performance of

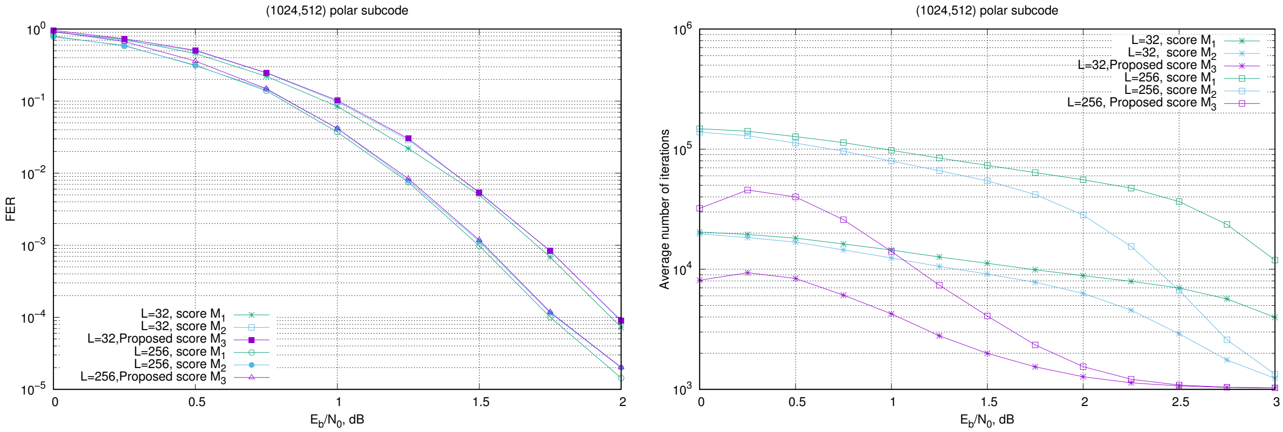

Fig. 1. The impact of the score function on the decoder performance and complexity.

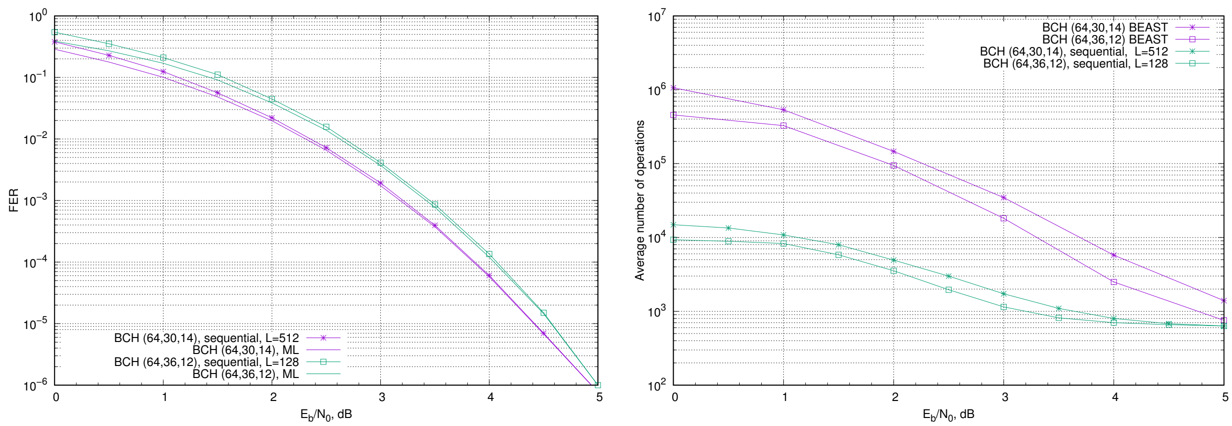

Fig. 2. Performance and complexity of extended BCH codes under sequential decoding

the decoder employing the proposed score M_3 is essentially the same as in the case of score M_2.

Table I provides the average number of arithmetic operations performed by the sequential decoder implementing the proposed path score function, and the one presented in [5]. It can be seen that employing the proposed score function results in substantially lower average decoding complexity.

The proposed approach can be also immediately used for decoding of short extended BCH codes by exploiting their representation via a dynamic freezing constraint matrix [3]. Figure 2 shows that the sequential algorithm provides near-ML decoding of these codes with substantially lower complexity compared to the BEAST algorithm, a simple ML decoding method for arbitrary linear block codes [12].

Acknowledgement

The author thanks Dr. V. Miloslavskaya for many fruitful discussions.

References

[1] E. Arikan, "Channel polarization: A method for constructing capacity-achieving codes for symmetric binary-input memoryless channels," *IEEE Trans. on Inf. Theory*, vol. 55, no. 7, pp. 3051–3073, July 2009.

[2] I. Tal and A. Vardy, "List decoding of polar codes," *IEEE Trans. On Inf. Theory*, vol. 61, no. 5, pp. 2213–2226, May 2015.

[3] P. Trifonov and V. Miloslavskaya, "Polar subcodes," *IEEE J. on Selected Areas in Communications*, vol. 34, no. 2, pp. 254–266, February 2016.

[4] K. Niu and K. Chen, "Stack decoding of polar codes," *Electronics Letters*, vol. 48, no. 12, pp. 695–697, June 2012.

[5] V. Miloslavskaya and P. Trifonov, "Sequential decoding of polar codes," *IEEE Communications Letters*, vol. 18, no. 7, pp. 1127–1130, 2014.

[6] K. S. Zigangirov, "Some sequential decoding procedures," *Problems of Information Transmission*, vol. 2, no. 4, pp. 1–10, 1966, in Russian.

[7] R. Johannesson and K. Zigangirov, *Fundamentals of Convolutional Coding*. IEEE Press, 1998.

[8] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to Algorithms*, 2nd ed. The MIT Press, 2001.

[9] J. Massey, "Variable-length codes and the Fano metric," *IEEE Trans. on Inf. Theory*, vol. 18, no. 1, pp. 196–198, January 1972.

[10] A. Balatsoukas-Stimming, M. B. Parizi, and A. Burg, "LLR-based successive cancellation list decoding of polar codes," *IEEE Transactions On Signal Processing*, vol. 63, no. 19, pp. 5165–5179, October 2015.

[11] D. Kern, S. Vorkoper, and V. Kuhn, "A new code construction for polar codes using min-sum density," in *Proc. of International Symposium on Turbo Codes and Iterative Information Processing*, 2014, pp. 228–232.

[12] I. E. Bocharova, R. Johannesson, B. D. Kudryashov, and M. Loncar, "BEAST decoding for block codes," *European Transactions on Telecommunnications*, vol. 15, no. 4, p. 297–305, July 2004.