Waterfall Region Performance of Punctured LDPC Codes over the BEC

Iryna Andriyanova  
ENSEA  
ETIS group  
95015 Cergy, France  
iryna.andriyanova@ensea.fr

Rudiger Urbanke  
EPFL, School of Communications & Computer Sciences  
1015 Lausanne, Switzerland  
rudiger.urbanke@epfl.ch

Abstract—This paper is devoted to the finite-length analysis of iterative performance of punctured LDPC ensembles in the waterfall region, assuming the transmission over the binary erasure channel (BEC). The analysis is carried out using the scaling approach proposed in [1]. Two punctured ensembles are considered: (a) randomly punctured ensembles, in the sense that each bit of a codeword is punctured with a puncturing probability; (b) ensembles with a fixed punctured fraction of bits of each degree. In both cases, parameters of the scaling approximation are completely determined in terms of the code ensemble parameters such as left, right and puncturing degree distributions.

I. INTRODUCTION

Puncturing is a technique playing an important role in rate-adaptable coding schemes, which, in their turn, are commonly used for transmission over time-varying channels. The advantage of rate-adaptable schemes is in the use of only one encoder/decoder pair at any transmission rate within the required range. The main idea of the technique is to not transmit, i.e. to puncture a part of bits in the codeword, the amount of which will depend on the channel state. If the channel is good enough, that a small number of code bits needs to be transmitted to be able to recover the whole codeword.

Puncturing was first proposed in [2]; since then a number of literature is available on the design of various classes of punctured (rate-compatible) codes. What concerns the decoding analysis, the asymptotic iterative decoding performance of punctured LDPC codes was studied in [3]. Also, the optimization of asymptotic puncturing degree distributions (i.e. a selection of punctured symbols) by means of linear programming was proposed. This work was continued in [4], where authors considered the design of puncturing patterns for short-length codes through defining the notion of the recoverability level. In [5], authors proved the existence of punctured ensembles with good minimum distance properties for an arbitrary range of rates. Further, average weight distributions of punctured LDPC codes were considered in [6]. It was shown that one achieves the capacity of a memoryless binary symmetric channel under ML decoding by puncturing a code ensemble of smaller rate. The design of linear-time encodable rate-compatible LDPC codes has been discussed recently in [7]. As for the iterative performance of finite-length punctured codes, some bounds on their average bit error rate for LDPC codes over the BEC were presented in [8]; these results have been obtained by bounding the average erasure probability at the output of the decoder, expressed in terms of the channel parameter, of the puncturing fraction and of stopping set weights. However, any result on the block error rate of punctured codes is still not known.

In this paper, we present new results related to the behavior of punctured LDPC code ensembles under iterative decoding. We consider a punctured LDPC ensemble of some finite codelength and approximate its block erasure rate by applying the scaling approach given in [1] and further developed in [9], [10]. We define two different punctured ensembles, so called randomly punctured LDPC codes and for fixed punctured LDPC codes, and perform the analysis for both of them. The paper is organized as follows. In Section II, two punctured LDPC ensembles are defined. Their asymptotic analysis is described in Section III. Section IV contains results related to the scaling approximation, namely the computation of its parameters α and β. Some results obtained by this approximation are compared with simulated performances in Section V. Section VI concludes the paper.

II. PUNCTURED LDPC CODES

An LDPC ensemble of length \( n \) is defined through an ensemble of bipartite graphs with two subsets of vertices, the subset of variable nodes and the subset of parity checks [11]. The ensemble is usually characterised by two polynomials, \( \lambda(x) = \sum_{i \in V} \lambda_i x^{i-1} \) and \( \rho(x) = \sum_{j \in C} \rho_j x^{j-1} \), where \( V \) (or \( C \)) is the set of degrees of variable nodes (or parity checks), and \( \lambda_i \) (\( \rho_j \)) is the fraction of edges connected to a variable node (to a parity check) of degree \( i \). Therefore, the corresponding LDPC code ensemble is denoted by \( \mathcal{C}(n, \lambda, \rho) \). Let us also define the polynomial \( \Lambda(x) = \sum_{i \in V} \Lambda_i x^{i-1} \), for which \( \Lambda_i \) represent the fraction of variable nodes of degree \( i, i \in V \).

We distinguish two puncturing schemes.

- In the first one, a code position corresponding to a variable node of degree \( i \) in the bipartite graph is punctured uniformly at random with some puncturing probability \( p_i \). Let us define the puncturing polynomial as \( p(x) = \sum_{i \in V} p_i x^{i-1} \). The randomly punctured code ensemble \( \mathcal{C}_R(n, \lambda, \rho, p) \) is obtained by taking into account all the possible realizations of puncturing patterns with the puncturing degree distribution \( p(x) \).

- For a fixed punctured ensemble \( \mathcal{C}_F(n, \lambda, \rho, p) \), the exact fraction \( p_i \) of code positions corresponding to variable nodes of degree \( i \) is punctured, \( i \in V \). With some abuse of notation, we denote the fixed puncturing polynomial by \( p(x) = \sum_{i \in V} p_i x^{i-1} \).
For $n \to \infty$, we denote the corresponding randomly and fixed punctured ensembles by $C_R(\lambda, \rho, p)$ and $C_F(\lambda, \rho, p)$.

Remark: Notice that the two schemes are versions of the “intentional puncturing” presented in [4], where code positions corresponding to different column weights have different punctured fractions. An alternative to the intentional puncturing is the so called “random puncturing” [4], where all the code bits are punctured in the same way. Clearly, for regular codes, these two notions are the same. However, in general, the concentration result [12] does not hold for ensembles obtained by random puncturing.

Through all the paper we assume the transmission over the BEC with channel erasure probability $\epsilon$, using randomly or fixed punctured LDPC codes. Therefore, the transmission scheme can be also modeled as the transmission through two parallel channels, where the unpunctured bits are transmitted through the BEC with erasure probability $\epsilon$, and the punctured bits - through the BEC with erasure probability $1 - \epsilon$. The iterative decoder of the punctured code utilizes the structure of the original code. Such a model allows to use puncturing in order to transmit at variable rates using one encoder and one decoder. Remind [4] that the design rate $r$ of a punctured ensemble with a puncturing degree distribution $p(x)$ is

$$r = \frac{R}{1 - \sum_{i \in V} \rho_i \lambda_i x_i},$$

where $R$ is the initial design rate of the unpunctured ensemble.

### III. Iterative Decoding Threshold

In this section, well-known results on the iterative thresholds of punctured codes are reminded. But first let us give some notation.

**Notation 1:** Assume transmission over the BEC. Then $\epsilon^*$ denotes the iterative threshold of a given unpunctured code ensemble. Moreover, $\epsilon^*_p$ is the iterative threshold of punctured ensembles (it is easy to show that for both $C_R(\lambda, \rho, p)$ and $C_F(\lambda, \rho, p)$ the threshold is the same).

**Lemma 3.1 (Iterative thresholds of punctured ensembles):** Consider $C_R(\lambda, \rho, p)$ or $C_F(\lambda, \rho, p)$ and let

$$\lambda_p(x) = \sum_{i \in V} p_i \lambda_i x_i^{i-1}, \quad \tilde{\lambda}_p(x) = \sum_{i \in V} (1 - p_i) \lambda_i x_i^{i-1},$$

for both, so that $\lambda(x) = \lambda_p(x) + \tilde{\lambda}_p(x)$. Then the density evolution equation is

$$F(x, \epsilon) = x - \lambda_p(1 - \rho(\bar{x})) - \epsilon \tilde{\lambda}_p(1 - \rho(\bar{x})) = 0,$$

with $\bar{x} = 1 - x$.

Let us illustrate Lemma 3.1 with some examples:

**Example** For punctured LDPC codes with $\lambda(x) = x^2$, $\rho(x) = x^5$ and the puncturing polynomial $p(x)$, we get

$$p(x) = \begin{array}{c|c|c|c|c} r & 0 & 0.5x & x & x + 0.1429x^3 \\ \hline \rho & 3/7 & 0.5 & 0.6 & 0.7 \\ \epsilon^*_p & 0.4828 & 0.42 & 0.3319 & 0.2206 \end{array}$$

### IV. Finite-Length Scaling Approximation

According to the general scaling hypothesis [13], one supposes that, in the waterfall region, the error probability under iterative decoding of almost any sparse-graph ensemble follows a scaling law. First results in this direction concerned LDPC codes [14], the approach has been generalized to turbo-like code ensembles later on [15].

In his work, we use a new derivation of the scaling law proposed in [16]. Following it, the average block error probability of a code ensemble $C(n, \lambda, \rho, p)$ is $P_T(n, \lambda, \rho, p) = Q(\frac{\sigma^2}{\sigma^*_R})$, where

$$\sigma^2 = \mathbb{E}[(\hat{H}_x - \mathbb{E}[\hat{H}_x])^2].$$

Here $\hat{H}_x$ is a random variable related to the fraction of erased bits at the input of the iterative decoder given that fraction of erased bit at its output is $x^*$. In the case the randomly punctured code ensemble $C_R(n, \lambda, \rho, p)$, a bit of degree $i$ is erased independently and uniformly at random with probability $p_i + (1 - p_i)\epsilon$. Therefore, the variance of the equivalent communication channel is $\sigma^2_R$ is the mix of binary erasure channels parametrized by $r_i$’s, i.e.

$$\sigma^2_R = \sum_i \lambda_i \sigma^2_{p_i} + (1 - p_i)\epsilon.$$

Now, consider the fixed punctured ensemble $C_F(\lambda, \rho, p)$. From the ensemble point of view, it can be shown that the fixed puncturing scenario is equivalent to the following one: during puncturing, a bit corresponding to a variable node of degree $i$ is punctured uniformly at random with probability $p_i$. Then, exactly $(1 - p_i)n$ bits of degree $i$ pass through the erasure channel with probability $\epsilon$ and the $p_in$ bits pass through an ideal channel with erasure probability $\bar{\epsilon}^2$. Finally, $p_in$ bits of degree $i$ can be seen as transmitted through the BEC with erasure probability $p_i + (1 - p_i)\epsilon$. The variance $\sigma^2_F$ of such equivalent channel can be shown to be equal to

$$\sigma^2_F = \bar{\rho}_i^2(\rho(1 - \bar{\rho}^2) + (1 - \bar{\rho})^2 \sigma^2_{p_i} + (1 - p_i)\epsilon^2 + \sum_i \Lambda_i (1 - \Lambda_i)(p_i^2 + (1 - p_i)\epsilon^2),$$

with $\bar{\rho} = \sum_i \Lambda_i p_i$. It is known from [14], [16] that $\sigma = \sqrt{\frac{\alpha}{n}}$, where $\alpha$ is the so called scaling parameter which depending only on ensemble parameters. Similarly, $\sigma_R = \sqrt{\frac{\alpha R}{n}}$. Then

1 both randomly and fixed punctured

2 assuming that $p_i n$ and $(1 - p_i)n$ are integers
from (5) one can express $\alpha_F$ through $\alpha_R$ as
\[
\alpha_F = [p^2 \{ \tilde{p}(1 - \tilde{p}) \} + \sum_i \Lambda_i (1 - \Lambda_i) \lambda^2_i + (1 - \lambda)^2 \alpha^2_R]^{1/2}.
\] (6)

Notice that in the case of a regular ensemble with punctured fraction $p$, the expression simply becomes
\[
\alpha_F = \sqrt{p^2 \{ p(1-p) \} + (1-p)^2 \alpha^2_R}.
\]

**Conjecture 4.1:** Suppose the transmission was made over the BEC with erasure probability $\epsilon$ using a code from a punctured LDPC ensemble, $C_R(n, \lambda, \rho, p)$ or $C_F(n, \lambda, \rho, p)$. Then the expected block erasure probability is approximated respectively by
\[
P_B(C_R, n, \epsilon) \approx Q \left( \frac{\sqrt{n} (\epsilon - \epsilon - \beta_R R n^{-2/3})}{\alpha_R} \right),
\]
(7)
\[
P_B(C_F, n, \epsilon) \approx Q \left( \frac{\sqrt{n} (\epsilon - \epsilon - \beta_F R n^{-2/3})}{\alpha_F} \right).
\]
(8)

In (7) and (8), $\epsilon_p^*$ is the iterative decoding threshold, $\alpha_R$ and $\alpha_F$ are scaling parameters, $\beta_R$ and $\beta_F$ are shift parameters of corresponding punctured ensembles.

### A. Scaling Parameter $\alpha_R$

Owing to the relation (6), we only need to compute $\alpha_R$ of the randomly punctured ensemble in order to determine the slope in the waterfall region for both ensembles. We use the expression from [16]:
\[
\alpha_R = \left. \frac{\partial^2 \epsilon_p(x)}{\partial x^2} \right|_{x=x_p^*} \lim_{\xi \rightarrow \epsilon_p^*} (x - x_p^*) \frac{\nu_p}{\Lambda(1)};
\]
(9)

here, $\epsilon_p(x)$ is implicitly defined by the density evolution equation (3), and $(\epsilon_p^*, x_p^*)$ is the (unique) critical point of (3). $\nu_p$ in (9) is the variance of erased messages for $\epsilon > \epsilon^*$, calculated in the limit of large codelengths when the number of iterations goes to infinity. $\nu$ can be written in the following form ([16])
\[
\nu_p = \frac{\xi_p}{(1 - \mu_p(x))^2},
\]
(10)

where $\mu_p(x)$ is the second largest eigenvalue of a matrix related to the density evolution process, and $\xi_p$ is some parameter depending only on $\epsilon_p^*, x_p^*$ and on code ensemble parameters $\lambda, \rho$ and $p$. In this case we obtain the expression for $\alpha_R$ which is similar to the expression of $\alpha$ for unpunctured ensembles, stated in the recent paper [9]. We have the following theorem:

**Theorem 4.1 ($\alpha_R$ of $C_R(\lambda, \rho, p)$):** Consider $C_R(\lambda, \rho, p)$ and let $\alpha_R$ be the scaling parameter of the ensemble. Define also
\[
y(x) = 1 - \rho'(\tilde{x}),
\]
(11)
\[
\pi(y) = \epsilon \lambda_p(y) + \lambda_p(y),
\]
(12)
\[
\mu_p(x) = \pi'(y) \rho'(\tilde{x}).
\]
(13)

Then
\[
\alpha_R = \frac{\pi'^*}{\lambda_p}(1 - 2 \lambda_p' \lambda_p(x)(1 - \mu_p)) \sqrt{(1 - y_p')(\rho'(1) - \rho'(\tilde{x}^*))}. \]  (14)

In this expression, we omit the dependence of $\pi, \mu_p$ and $\lambda_p$ from $x_p^*, y_p^*$ and $\epsilon_p^*$ for short.

**Proof:** To compute $\alpha_R$, we proceed exactly as it has been done for the unpunctured case in [16]. We obtain the desired expression (14) by substituting parts in (9) and (10) by
\[
\lim_{\xi \rightarrow \epsilon_p^*} \frac{x - x_p^*}{1 - \mu_p(x)} = \frac{1}{1 - \hat{\mu}'(\tilde{x}^*)},
\]
\[
\frac{\partial^2 \epsilon}{\partial x^2} = - \frac{\mu_p(x)}{\lambda_p(y)} + \frac{2 \lambda_p(y) \rho'(\tilde{x})}{\lambda_p(y)^2}.
\]

### B. Shift Parameters $\beta_R$ and $\beta_F$

We will derive shift parameters $\beta_R$ and $\beta_F$ by following the derivation of $\beta$ in the unpunctured case, given in [10], [17]. In more detail, we will define an appropriate branching process for both punctured ensembles and determine statistics of random variables related to it.

Choose at random a code from a punctured LDPC ensemble of infinite codelength ($C_R(\lambda, \rho, p)$ or $C_F(\lambda, \rho, p)$) and consider its corresponding bipartite graph. Assume the transmission over the BEC with erasure probability $\epsilon_p^*$ and let $y$ be the realization of the channel output. Pick an unpunctured variable node $v$ such that its corresponding bit value has been erased during the transmission. Form another channel realization $\bar{y}$ by changing the channel output for this code bit, i.e., by letting it to be unerased. Hence we obtain two channel realizations $y$ and $\bar{y}$ different in only one position, and now we perform the decoding for both of them. Consider the computation tree of $v$ for both cases. Due to infinite length assumption, it does not contain any cycles. The change of the initial message of $v$ from erased to known will imply the difference in messages at some variable nodes in the computation tree computed. Thus we construct a branching process with the following representation tree: its root correspond to the variable node $v$, and its vertices correspond to the variable nodes whose messages differ for two channel realizations $y$ and $\bar{y}$. Denote by $Z_{t+1}$ the number of changes in the messages at variable nodes at depth $t + 1$ from the root node. Then
\[
Z_{t+1} = \sum_{i=1}^{Z_t} X_p,
\]
where $X_p$ is distributed as the number of changes in messages of variable nodes in the representation tree. The initial distribution $Z_1$ is denoted by $W_p$, i.e. $W_p$ is a random variable distributed as the number of changes in the messages of variable nodes adjacent with the initial variable node $v$.

We also define $S_p = \lim_{\epsilon \rightarrow \epsilon_p^*} \sqrt{\epsilon - \epsilon_p^*(1 - \mu_p(x))}$. To compute $\beta_R$ and $\beta_F$, we use following expressions,
similar to the one developed in the unpunctured case in [10].

\[ \beta_R = \left( \frac{\sqrt{2} S_R Var[X_R]}{E[W_R]} \right)^{2/3}, \quad (15) \]

\[ \beta_F = \left( \frac{\sqrt{2} S_F Var[X_F]}{E[W_F]} \right)^{2/3}, \quad (16) \]

and prove four following lemmas.

**Lemma 4.1:** \( S_R = S_F = S \), with

\[ S = (2\lambda_p(y_p) - \pi''(y_p)\rho'(x_p)\pi''(x_p))^{1/2}. \quad (17) \]

**Proof:** For both \( C_R(\lambda, \rho, p) \) and \( C_F(\lambda, \rho, p) \),

\[ \lim_{\epsilon_p \to \epsilon_p^+} \frac{F'_p}{1 - \mu_p(x)} = \lim_{\epsilon_p \to \epsilon_p^-} 2F'_p \cdot (\mu'_p(x) \cdot F'_p)^{1/2} \]

\[ = (2F'_p \cdot \mu'_p(x))^{1/2} \]

\[ \frac{1}{\sqrt{-2\lambda_p(y_p)\pi'(y_p)^2(1 - x)}} \]

where \( F(x, c) = 0 \) is the density evolution equation.

**Lemma 4.2:**

\[ E[W_R] = E[W_F] = \lambda_p(y_p)\rho'(x_p). \quad (18) \]

The proof of the lemma follows exactly the lines of the similar proof given in [17], for both punctured ensembles.

Let us compute \( Var[X_R] \) and \( Var[X_F] \).

**Lemma 4.3:** For \( C_F(\lambda, \rho, p) \),

\[ Var[X_F] = \frac{\lambda(y_p)\rho(x_p)}{\lambda(y_p)} \left( 1 - \frac{x_p\rho'(x_p)}{y_p} \right) \]

\[ + \frac{x_p\rho'(x_p)^2}{y_p} \left( \pi'(y_p) + \frac{\pi'(y_p) - \pi'(y_p^2)}{y_p - x_p} \right). \quad (19) \]

**Proof:** Again we follow the reasoning presented in [17]. Let \( L \) be the number of variable nodes which changed their messages. Then

\[ X_F = \sum_{i=1}^{L} W_i, \]

where \( W \) is a random variable distributed as the number of offsprings of one parent on the branching process. \( W \) follows the Bernoulli distribution with parameter \( z = \frac{x_p\rho(x_p)}{y_p} \).

Therefore, the mean and the variance of the distribution are

\[ E[W] = z, \quad Var[W] = z(1 - z) \quad (20) \]

Now we need to compute the mean and the variance of the distribution of \( L \). We know that

\[ P(L = l) = \frac{\lambda_p(y_p)^l \cdot \pi_{l+1}(1 - p_l)(y_p^2)^l}{\pi(y_p)}, \quad (21) \]

where \( p_l \)’s are fixed puncturing fractions. Therefore,

\[ E[L] = \frac{y_p\pi'(y_p^2)}{\pi(y_p)}, \quad (22) \]

\[ Var[L] = \frac{(y_p^2\pi''(y_p^2) + y_p\pi'(y_p^2)) - \pi'(y_p^2)^2}{\pi(y_p)^2}. \quad (23) \]

We obtain the variance for \( X_F \):


\[ = \frac{x_p\rho'(x_p)^2}{y_p} \left( \pi'(y_p) + \frac{\pi'(y_p) - \pi'(y_p^2)}{y_p - x_p} \right) \]

\[ + (1 - \pi_p^2) \sum_{l} lp_l(1 - p_l)\lambda_{l+1}(y_p^2)^{-l} \cdot \pi(y_p). \quad (24) \]

**Lemma 4.4:** For \( C_R(\lambda, \rho, p) \),

\[ Var[X_R] = Var[X_F] + \frac{x_p\rho'(x_p)^2}{y_p} \left( \pi'(y_p^2) + \frac{\pi'(y_p) - \pi'(y_p^2)}{y_p - x_p} \right) \]

\[ + (1 - \pi_p^2) \sum_{l} lp_l(1 - p_l)\lambda_{l+1}(y_p^2)^{-l} \cdot \pi(y_p). \quad (25) \]

**Proof:** The proof is similar to the previous one with only one exception: for \( C_R(\lambda, \rho, p) \), \( p_l \)’s of Eq(21) represent probabilities. Therefore, by the low of total variance,

\[ Var[L] = \frac{(y_p^2\pi''(y_p^2) + y_p\pi'(y_p^2)) - \pi'(y_p^2)^2}{\pi(y_p)^2} \]

\[ + (1 - \pi_p^2) \sum_{l} lp_l(1 - p_l)\lambda_{l+1}(y_p^2)^{-l} \cdot \pi(y_p). \quad (26) \]

The final expression for \( Var[X_R] \) is given by (25).

**V. Numerical Examples**

We begin by comparing the average performance of two punctured ensembles in Fig. 1. Here red curves correspond to a randomly punctured ensemble and blue ones - to a fixed punctured one. Their scaling estimations are given respectively by black and magenta curves. Both code ensembles have \( \lambda(x) = x^2 \) and \( \rho(x) = x^5 \); the codeweight is equal to 1024. One can note that the ensemble \( C_F(x^2, x^5, p) \) is performing better than \( C_R(x^2, x^5, p) \). This is justified by the fact that \( \sigma^2_F \leq \sigma^2_R \). One can also see that the presented estimations match well with the simulated curves.

![Fig. 1. Average performance of \( C_R(x^2, x^5, p) \) (in red) and of \( C_F(x^2, x^5, p) \) (in blue) for \( p = 0.1 \) (on the right) and \( p = 0.2 \) (on the left). Black and magenta curves are corresponding scaling estimations.](image-url)
Next two examples concern fixed punctured LDPC codes. Consider the regular ensemble \( C_F(x^2, x^5, p) \). Its scaling and shift parameters are represented in the table below. Also, the obtained scaling approximation is compared with the simulated performance in Fig.2.

<table>
<thead>
<tr>
<th>( p )</th>
<th>( \alpha_F )</th>
<th>( \beta_F )</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0.5608</td>
<td>0.6171</td>
</tr>
<tr>
<td>0.1</td>
<td>0.5616</td>
<td>0.6857</td>
</tr>
<tr>
<td>0.2</td>
<td>0.5605</td>
<td>0.7714</td>
</tr>
</tbody>
</table>

From Fig.2, one can see that the behaviour of the word-error rate of the ensemble is accurately predicted by scaling approach for all chosen codelengths.

To give an example of irregular ensembles, consider \( C_F(x^6 + 5x^5, x^5) \) as initial unpunctured ensemble and two fixed puncturing degree distributions, \( p(x) = 0.5x \) and \( p(x) = x \). We obtain following values of \( \alpha_F \) and \( \beta_F \):

<table>
<thead>
<tr>
<th>( p(x) )</th>
<th>( \alpha_F )</th>
<th>( \beta_F )</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0.5568</td>
<td>0.6996</td>
</tr>
<tr>
<td>0.5x</td>
<td>0.6255</td>
<td>0.8395</td>
</tr>
<tr>
<td>x</td>
<td>0.7442</td>
<td>1.2627</td>
</tr>
</tbody>
</table>

Simulated and estimated performances of these punctured ensembles are presented in Fig.3.

VI. CONCLUSION

We have proposed a finite-length approximation of two punctured LDPC ensembles: so called randomly and fixed punctured ensembles. Our approximation is based on the scaling approach. We have derived explicit expressions of its parameters by using known results for the unpunctured case.

For simplicity, the transmission was assumed to take place over the BEC. The presented result can be generalized to other binary memoryless symmetric channels, namely to the Gaussian channel, and it might be applied to design the puncturing patterns for finite-length rate-adaptable coding schemes.

VII. ACKNOWLEDGEMENT

Authors thank Sewoong Oh for the preprint of [17].

Fig. 2. Comparison of the finite-length approximation for \( C_F(x^2, x^5, p) \) with simulations. The puncturing fraction \( p \) is equal to 0.2 for the curves on the left, 0.1 - in the middle and 0 on the left. Simulated code lengths are 1024, 2048, 4096 and 8192. Black lines represent simulated curves, red and blue - finite-length estimations.

Fig. 3. Comparison of the finite-length approximation for \( C_F(\frac{x^2}{6} + 2x^5, p(x)) \) with simulations. \( p(x) = x \) for the bunch of curve on the left, \( p(x) = 0.5x \) for those in the middle and \( p(x) = 0 \) for curves on the right. Chosen code lengths are 1022 and 2051. Black lines represent finite-length estimations and red lines - numerical results.

REFERENCES