Probabilistic Shaping for Asymmetric Channels and Low-Density Parity-Check Codes

Thomas Wiegart*, Linfang Wang†, Diego Lentner*, Richard D. Wesel†
*Institute for Communications Engineering, Technical University of Munich, Germany
†Communications Systems Laboratory, University of California Los Angeles, USA
emails: thomas.wiegart@tum.de, lfwang@g.ucla.edu, diego.lentner@tum.de, wesel@ucla.edu

***Abstract*—An algorithm is proposed to encode low-density parity-check (LDPC) codes into codewords with a non-uniform distribution. This enables power-efficient signalling for asymmetric channels. We show gains of $0.9\,\mathrm{dB}$ for additive white Gaussian noise (AWGN) channels with on-off keying modulation using 5G LDPC codes.**

***Index Terms*—LDPC codes, probabilistic shaping, forward-error correction, asymmetric signalling.**

I. Introduction

This paper explores low-density parity-check (LDPC) coded communications over asymmetric channels such as the additive white Gaussian noise (AWGN) channel with on-off keying (OOK). To approach capacity, one must usually use shaping so that signal points are not equally likely (probabilistic shaping), not uniformly spaced (geometric shaping), or both [1]–[5].

A popular technique called probabilistic amplitude shaping (PAS) [6], [7] uses a distribution matcher (DM) to perform shaping before the forward error correction (FEC) encoder, e.g., a constant composition DM (CCDM) [8]. PAS factors the target channel-input distribution as $P_X = P_S \cdot P_A$ where P_S is the uniform distribution used for parity bits and P_A relates to the input amplitudes. PAS can approach capacity for symmetric channels where the capacity-achieving P_X is symmetric. An especially attractive feature of PAS is its flexibility: PAS can be combined with any (systematic) FEC code, it permits fine rate adaptation, and it performs well with long [7] and short [9] block lengths.

PAS does not directly apply to channels where the optimal input distribution is asymmetric. For example, for the AWGN channel with OOK modulation, the authors of [10] propose a time sharing (TS) scheme that combines FEC with a non-uniform OOK signaling. In the TS scheme, a DM generates non-uniform bits from a uniform message, and a systematic FEC encoder appends uniformly-distributed parity bits.

Another probabilistic shaping scheme for OOK is presented in [11] that uses the method by Honda and Yamamoto [12]. Here, polar codes perform joint distribution matching and FEC by using a polar decoder to encode message bits into a subset of the code with the desired distribution. The polar coding scheme (asymptotically) generates OOK symbols with the capacity-achieving distribution and performs better than the TS scheme. It is, however, tied to the use of polar codes.

Several schemes have been proposed to shape LDPC codes non-uniformly by using a decoder to encode; see [13], [14]. However, these schemes may encode to an invalid codeword under belief propagation (BP) encoding. The papers [15], [16] report that spatial coupling and guided decimation can combat this problem, and the paper [14] proposes an outer FEC code to correct encoding errors. These workarounds do not guarantee a valid encoding.

The paper [17] proposes linear layered probabilistic shaping (LLPS) that shapes the codewords of a linear block code with two DMs: The message bits are shaped with a conventional DM and encoded systematically. A syndrome distribution matcher (SDM) shapes the parity bits by reserving ℓ bits and determining them to achieve a non-uniformly distributed codeword. However, the SDM requires enumerating all 2^ℓ combinations of these ℓ bits which is infeasible for large ℓ.

This paper proposes an efficient algorithm to approximate LLPS for LDPC codes. The algorithm uses systematic encoding and sequentially determines ℓ bits of the systematic part of the codeword. It it based on a BP-like algorithm on the Tanner graph of the generator matrix. The algorithm is sub-optimal but it has several attractive features: it generates a valid codeword, it can be implemented efficiently, and it provides reasonable power gains.

This paper is organized as follows: Section II introduces the AWGN channel with OOK modulation, LDPC codes, and LLPS. Section III presents the proposed algorithm and Section IV provides simulation results. Section V concludes the paper.

II. Preliminaries

A. Channel Model

Consider the AWGN channel with OOK modulation:

$$Y = X + N \tag{1}$$

where the transmit symbol X has alphabet $\{0, A\}$ and N is an additive white Gaussian noise term with zero mean and variance σ^2. We write $p_0 = P_X(0)$ and the signal-to-noise ratio (SNR) is

$$\gamma = \frac{(1-p_0)A^2}{\sigma^2}. \tag{2}$$

This work was supported by the German Research Foundation (DFG) through projects 390777439 and 509917421 and by the National Science Foundation (NSF) grant 1911166. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the NSF.

A non-uniform distribution P_X for X can be beneficial under an average transmit power constraint such as $\mathrm{E}[X^2] \leq P$. We map codeword symbol 0 to the 0-symbol and codeword symbol 1 to the A-symbol. Although this paper focuses on the AWGN channel with OOK modulation, the scheme can be applied to any binary-input asymmetric channel.

B. LDPC Codes

LDPC codes [1] are linear block codes with a sparse parity check matrix. They can be represented by a bipartite graph, called a Tanner graph, with variable nodes (VNs) representing the codeword symbols and check nodes (CNs) representing the parity checks. LDPC codes can be efficiently decoded using BP that iteratively exchanges likelihoods of the codeword bits between the check and variable nodes. Let $L_{\mathrm{V}_i \to \mathrm{C}_j}$ be the message from VN i to CN j and $L_{\mathrm{C}_j \to \mathrm{V}_i}$ be the message from CN j to VN i. The message update rules at the variable and check nodes are

$$L_{\mathrm{V}_i \to \mathrm{C}_j} = L_i + \sum_{k \in \mathcal{N}(\mathrm{V}_i), k \neq j} L_{\mathrm{C}_k \to \mathrm{V}_i} \tag{3}$$

$$L_{\mathrm{C}_j \to \mathrm{V}_i} = 2 \tanh^{-1} \left(\prod_{k \in \mathcal{N}(\mathrm{C}_j), k \neq i} \tanh \left(L_{\mathrm{V}_k \to \mathrm{C}_j} / 2 \right) \right) \tag{4}$$

where L_i is the log-likelihood ratio (LLR) of the code bit associated with VN i and $\mathcal{N}(\mathrm{V}_i)$ is the index-set of all neighbours of VN i, and similarly for $\mathcal{N}(\mathrm{C}_j)$. The a posteriori probability (APP) LLR of VN i is

$$L_i^{\mathrm{APP}} = L_i + \sum_{k \in \mathcal{N}(\mathrm{V}_i)} L_{\mathrm{C}_k \to \mathrm{V}_i} \tag{5}$$

and used to decide whether code bit i is zero ($L_i^{\mathrm{APP}} \geq 0$) or one ($L_i^{\mathrm{APP}} < 0$). The BP decoder is usually terminated once a valid codeword is found or one reaches a maximum number of iterations.

C. Linear Layered Probabilistic Shaping (LLPS)

LLPS [17] is an architecture to encode any linear code of length n_c and dimension k_c to non-uniform codewords. The output of a DM is encoded systematically. To obtain parity bits with non-uniform distribution with binary entropy $H_2(p_0)$, one needs to determine approximately

$$\ell \approx (n_c - k_c) \left(\frac{1}{H_2(p_0)} - 1 \right) \tag{6}$$

bits using a SDM.

Consider systematic encoding of a vector $\boldsymbol{v}$ of $k_c - \ell$ bits output by a DM. The parity check matrix $\boldsymbol{H}$ is partitioned as $\boldsymbol{H} = [\boldsymbol{H}_{\mathrm{s}} | \boldsymbol{H}_{\mathrm{p}}]$ where $\boldsymbol{H}_{\mathrm{s}}$ is a $m \times (k_c - \ell)$ matrix and $\boldsymbol{H}_{\mathrm{p}}$ is a full-rank $m \times (m + \ell)$ matrix with $m = n_c - k_c$. Next, calculate the syndrome $\boldsymbol{\eta} = \boldsymbol{v} \boldsymbol{H}_{\mathrm{s}}^T$ and observe that any $\boldsymbol{c} = [\boldsymbol{v} | \boldsymbol{p}]$ is a valid codeword if $\boldsymbol{p}$ is chosen such that $\boldsymbol{p} \boldsymbol{H}_{\mathrm{p}}^T = \boldsymbol{\eta}$ is fulfilled. There are 2^ℓ valid solutions for $\boldsymbol{p}$ and one can, e.g., choose the one with lowest Hamming weight $\mathrm{w}_{\mathrm{H}}(\boldsymbol{p})$:

$$\boldsymbol{p} = \underset{\boldsymbol{p}' \in \{0,1\}^{m+\ell}}{\operatorname{argmin}} \mathrm{w}_{\mathrm{H}}(\boldsymbol{p}') \quad \text{s.t.} \quad \boldsymbol{p}' \boldsymbol{H}_{\mathrm{p}}^T = \boldsymbol{\eta}. \tag{7}$$

The authors of [17] further decompose $\boldsymbol{H}_{\mathrm{p}}$ to solve (7) by enumerating over ℓ bits. This allows to shape a linear code with a desired distribution but has high complexity since 2^ℓ possibilities must be enumerated to solve (7). LLPS with SDM is thus feasible only for high-rate distribution matching with small ℓ (close-to-uniform distributions) and short blocklengths.

III. Shaped LDPC Codes

A. Principles of Shaped LDPC Codes

We propose an efficient BP-like algorithm for LLPS that approximates the solution of (7). Consider an (n_c, k_c) LDPC code with parity-check matix $\boldsymbol{H}$ and systematic generator matrix $\boldsymbol{G}_{\mathrm{sys}} = [\boldsymbol{I}_{k_c} \,|\, \boldsymbol{G}_{\mathrm{p}}]$ where $\boldsymbol{I}_{k_c}$ is the $k_c \times k_c$ identity matrix and $\boldsymbol{G}_{\mathrm{p}}$ is a $k_c \times (n_c - k_c)$ matrix. The codeword $\boldsymbol{c}$ is related to the input $\boldsymbol{u}$ by

$$\boldsymbol{c} = \boldsymbol{u}\boldsymbol{G} = [\boldsymbol{u} \,|\, \boldsymbol{u}\boldsymbol{G}_{\mathrm{p}}]. \tag{8}$$

We reserve ℓ of the k_c systematic bits for shaping. These shaping bits are a function of the remaining $k_c - \ell$ systematic bits and chosen such that the parity bits $\boldsymbol{u}\boldsymbol{G}_{\mathrm{p}}$ have a non-uniform distribution. The encoding procedure to determine the shaping bits is described in the next subsection.

Since we determine bits from the systematic part of the code, any choice of the shaping bits gives a valid codeword. The remaining $k_c - \ell$ systematic bits are assumed to be the output of a DM (we use the CCDM [8]) with rate $R_{\mathrm{DM}} = k/(k_c - \ell)$, where k is the length of the uniform information sequence to be transmitted. The overall transmission rate is

$$R = k/n_c = R_{\mathrm{DM}}(k_c - \ell)/n_c = R_{\mathrm{DM}} R_{\mathrm{c}} - \ell/n_c.$$

where $R_c = k_c/n_c$ is the code rate of the LDPC code.

B. Encoding

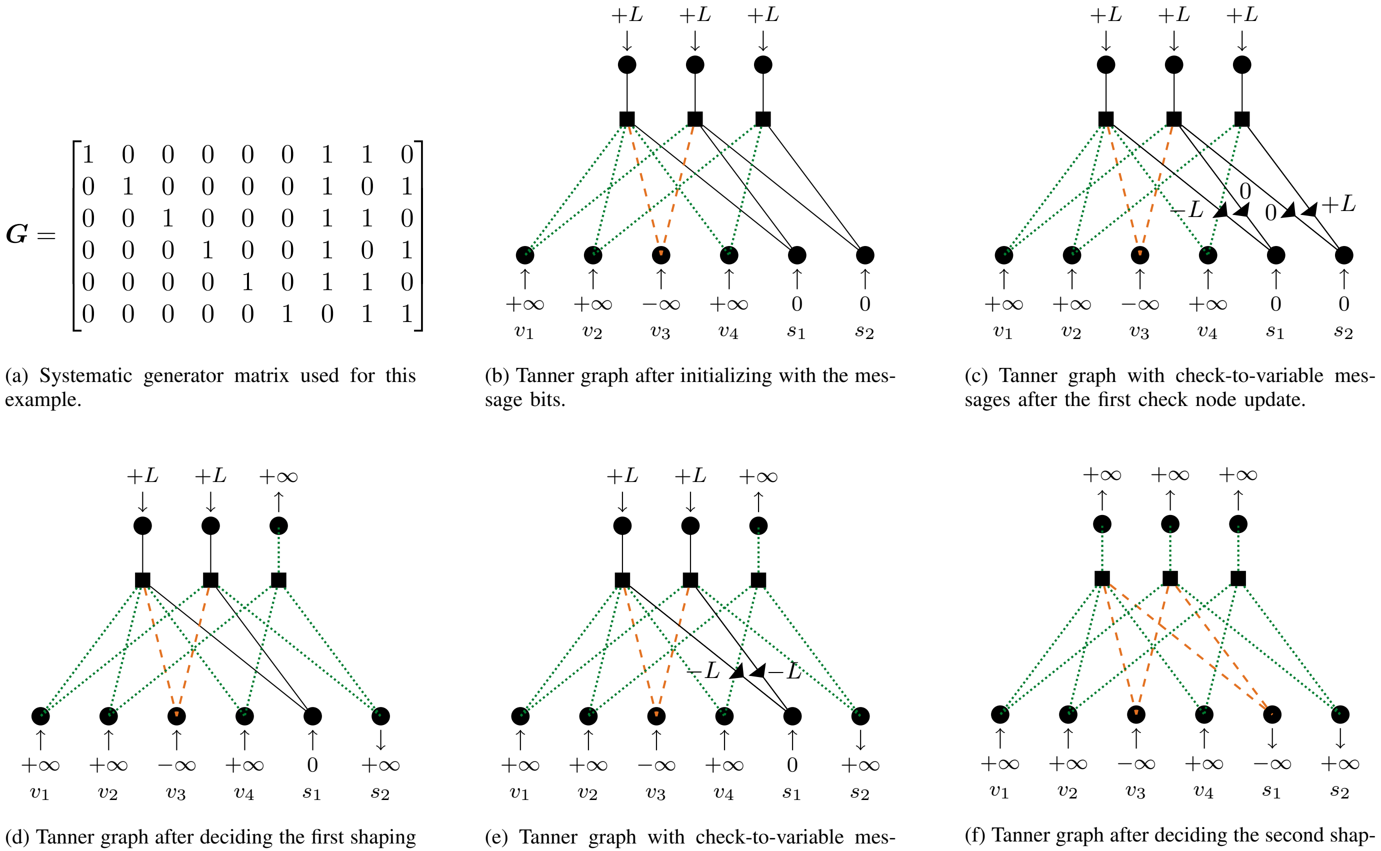
The algorithm encodes on the Tanner graph of the generator matrix using a BP-like algorithm that successively determines the ℓ shaping bits using ℓ iterations over the graph, i.e., each iteration determines one shaping bit. Fig. 1 shows an example with $n_c = 9$, $k_c = 6$, and $\ell = 2$.

The VNs of the Tanner graph are divided into three groups:

- *Message VNs:* the $k_c - \ell$ systematic VNs of the DM output $\boldsymbol{v}$ are fixed by setting the LLR L_i of VN i to $+\infty$ or $-\infty$ for a DM output 0 or 1, respectively.
- *Parity VNs:* the $n_c - k_c$ VNs of the parity bits have LLR $L_i = \log(p_0/(1 - p_0)) \triangleq +L$, which corresponds to the target distribution.
- *Shaping VNs:* the ℓ systematic VNs of the shaping bits are initialized with the LLR value $L_i = 0$. When making the final decision on these bits in (9), we will add the same offset $+L$ as for the parity VNs in order to induce the target distribution on the shaping bits as well.

Fig. 1b depicts the systematic message and shaping VNs on the bottom and the parity VNs on the top.

In the example, the LLRs of the message VNs are initialized according to the DM output $\boldsymbol{v} = [0, 0, 1, 0]$. All check-to-variable node messages are initialized with 0 and all variable-to-check node messages are initialized with the LLR of the



$$\boldsymbol{G} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 \end{bmatrix}$$

(a) Systematic generator matrix used for this example.

(b) Tanner graph after initializing with the message bits.

(c) Tanner graph with check-to-variable messages after the first check node update.

(d) Tanner graph after deciding the first shaping bit.

(e) Tanner graph with check-to-variable messages after the second check node update.

(f) Tanner graph after deciding the second shaping bit.

Fig. 1: Example of the encoding procedure with $n_c = 9$, $k_c = 6$, $\ell = 2$, and DM output $\boldsymbol{v} = [0, 0, 1, 0]$. The subfigures show (a) the generator matrix of the code and (b)-(f) the steps of the proposed encoding algorithm.

variable node. Thus, all messages from the message VNs are $\pm\infty$, all messages from the shaping VNs are 0, and all messages from the parity VNs are $+L$. Edges that carry messages from fixed VNs are marked as green-dotted ($L_i = +\infty$) or red-dashed ($L_i = -\infty$) in Fig. 1. These edges do not need to be updated and have a limited impact in the check node operations.

The algorithm now performs ℓ iterations on the graph to determine the ℓ shaping bits. Each iteration consists of the following steps.

- *CN updates:* update the check-to-variable messages to via (4). The update rule has three possible outcomes: if a CN has more than one edges to undetermined shaping VNs, all messages are 0. If a CN has exactly one edge to an undetermined shaping VN, then the message on this edge will be $+L$ or $-L$, depending on whether the number of incoming $-\infty$-message is even or odd; see Fig. 1c.
- *Determine one shaping bit (decimation step):* for all undetermined shaping VNs i, calculate

$$\widetilde{L}_i^{\text{APP}} = L_i^{\text{APP}} + L \tag{9}$$

with L_i^{APP} from (5). Choose one of the VNs with largest $|\widetilde{L}_i^{\text{APP}}|$. Fix the corresponding shaping bit using

$$s_i = \begin{cases} 0, & \widetilde{L}_i^{\text{APP}} \geq 0 \\ 1, & \text{else} \end{cases} \tag{10}$$

and change the LLR L_i of VN i accordingly. Further, update the messages on the edges connected to this VN. We remark that the $\widetilde{L}_i^{\text{APP}}$ are always integer multiples of L and can be interpreted as how many parity bits are (in this iteration) set to zero compared to how many are set to one. The offset $+L$ in $\widetilde{L}_i^{\text{APP}}$ is then applied to bias the shaping bits towards the non-uniform target distribution. In the example, we have $\widetilde{L}_1^{\text{APP}} = 0$ and $\widetilde{L}_2^{\text{APP}} = +2L$ and thus fix $s_2 = 0$. The updated graph is shown in Fig. 1d. Observe that fixing $s_2 = 0$ determines the third parity bit to be 0, as illustrated by the outgoing APP LLR of $+\infty$.

After ℓ iterations, the algorithm determined all ℓ shaping bits, and in a final stage it calculates all parity bits and puts out a valid codeword with a non-uniform distribution.

C. Decoding

Since all shaping bits are in the systematic part of the codeword we need not modify the decoder. After decoding, one simply discards the shaping bits. We remark that the shaping bits could be used for error detection by re-encoding

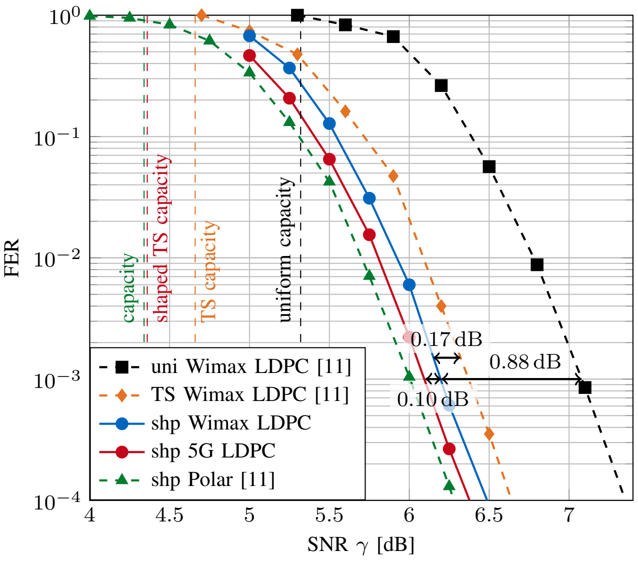


Fig. 2: FER with overall rate $R = 2/3$ and $n_c \approx 1000$ bits. The uniform (- ■-) and TS (- ◆-) Wimax LDPC codes have $n_c = 1056$ and are taken from [11]. The TS scheme has code rate $R_c = 0.75$. The shaped Wimax LDPC code (—●—) has $n_c = 1056$, $R_c = 0.75$, $\ell = 10$ shaping bits, and achieves an empirical distribution $p_0 = 0.65$. The shaped 5G LDPC code (—●—) has $n_c = 1008$, $R_c = 0.78$, 72 punctured bits, $\ell = 32$, and $p_0 = 0.69$. The shaped polar code (- ▲-) is from [11] and has $n_c = 1024$ and $p_0 = 0.69$.

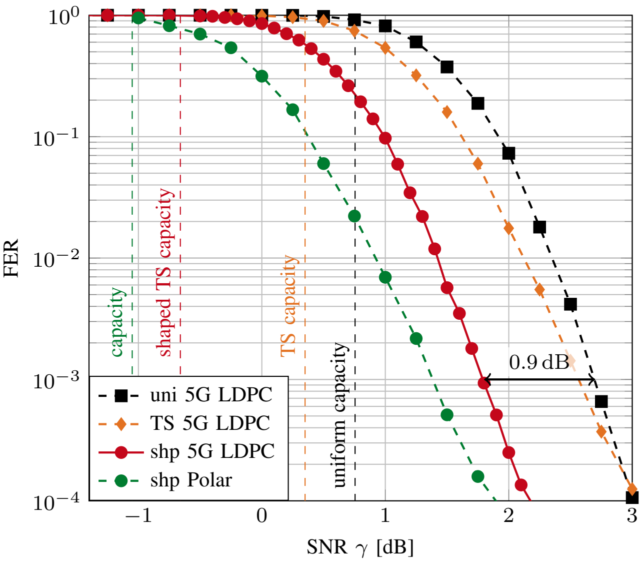


Fig. 3: FER with overall rate $R = 1/3$ and $n_c \approx 1000$ bits. The LDPC codes are from the 5G standard and have $n_c = 1056$. The TS LDPC code (- ◆-) has $R_c = 1/2$ (according to [10, Table I]). The shaped LDPC code (—●—) has $R_c = 2/3$, 64 punctured bits, $\ell = 64$ shaping bits, and empirical $p_0 = 0.80$. 3 The shaped polar code (- ●-) has $n_c = 1024$ and uses both SCL encoding and decoding with $L_{\text{SCL}} = 32$ and has $p_0 = 0.83$.

the message and comparing the decoded shaping bits with the re-encoded ones as in [18].

D. Discussion

The algorithm is based on the Tanner graph of the systematic generator matrix instead of the parity check matrix. The former matrix is usually denser than the latter. From the perspective of computational complexity this is not an issue as the CN update (4) only needs to be calculated if the effective CN degree is 1. However, if all check-to-variable node messages are 0, then the algorithm must guess one bit. The algorithm might thus perform poorly if the generator matrix is excessively dense or if one approximates an extreme distribution with many shaping bits. The behaviour improves by optimizing the decimation strategy and the positions of the shaping bits, by designing tailored codes, or by using more complex decimation strategies. For example, one may choose multiple shaping bits jointly or use brute-force search for the first bits and encode the remaining bits for multiple choices.

IV. Simulation Results

We simulated performance over an AWGN channel with OOK modulation, as described in Sec. II-A. We compare our scheme to the polar codes from [11] and the TS scheme from [10] with LDPC codes.

For some curves we used 5G LDPC codes [19]. They are all obtained from base graph 1. According to the standard, some of the systematic bits are punctured. We place the shaping bits within the punctured bits and slightly modify (9) to $\widetilde{L}_i^{\text{APP}} = L_i^{\text{APP}}$, i.e., we discard the additional offset of $+L$ as the punctured bits do not need to have a non-uniform distribution. If there are more punctured bits than shaping bits, the remaining punctured bits are filled with information bits.

Fig. 2 considers the transmission rate $R = 2/3$ and compares the frame error rates (FERs) of the proposed algorithm to uniform signaling, the TS scheme from [10], and shaped polar codes. With LDPC codes from the Wimax standard [20] we gain approximately 0.88 dB compared to uniform signaling and 0.17 dB compared to the TS scheme. For the shaped and TS curves we used $R_c = 0.75$; the shaped curve was generated using $\ell = 10$ shaping bits. We can further improve by using a 5G LDPC code. We chose a code with $n_c = 1008$ (72 punctured bits) and used $\ell = 32$. This gives an additional 0.1 dB of gain compared to the shaped Wimax LDPC code. We further depict the shaped polar code from [11, Fig. 5] for reference. The polar code has $n_c = 1024$, is combined with an outer 16 bit CRC, and a successive cancellation list (SCL) decoder [21] with list size $L_{\text{SCL}} = 32$ was used for encoding and for decoding. Both schemes perform very close to each other, but the polar code performs slightly better. This is expected for such rather short block lengths and we remark that the polar code has a tailored and optimized design, while the off-the-shelf LDPC code was not designed for this purpose.

The figure also depicts achievable rates as vertical dashed lines. As the distribution of the systematic part and the distribution of the parity part are not necessarily the same, our scheme also works in a TS manner. We depict the achievable rate with the empirical distributions created by our algorithm as “shaped TS capacity”. The “TS capacity” line corresponds to the achievable rate with the TS scheme from [10]. For this scenario, the achievable rate of our scheme is very close to

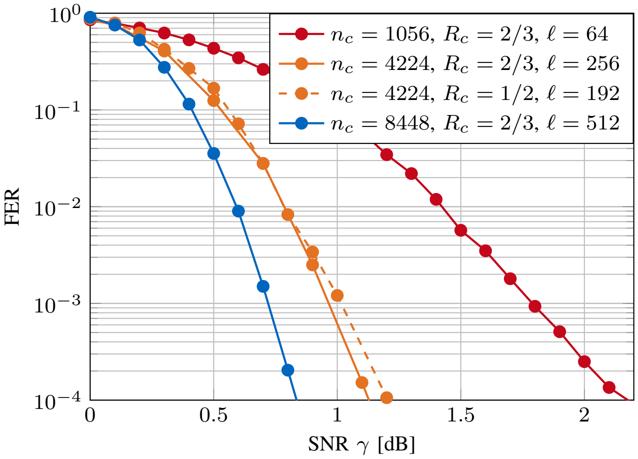


Fig. 4: FER with overall rate $R = 1/3$ of shaped 5G LDPC codes with different code lengths and code rates. All curves with $R_c = 2/3$ have empirical $p_0 = 0.80$ and the curve with $R_c = 1/2$ has $p_0 = 0.71$.

capacity and the gains observed in the FER curves match the theoretical predictions.

Fig. 3 shows results for a transmission rate of $R = 1/3$. Here, the theoretical shaping gain is $1.8\,\text{dB}$. The shaping gain of the TS scheme is small due to the high number of uniform parity bits for low transmission rates [10]. The proposed scheme with a 5G code (64 punctured bits and $\ell = 64$) gains $0.9\,\text{dB}$ compared to uniform transmission. This is a bit less then it could be expected from the achievable rates (and we also observe that the FER waterfall is not accurately predicted by the capacity for the shaped curve). We believe that this is a code design issue and further gains are possible by designing tailored codes. We remark that the target distribution for this scenario is quite biased ($p_0 = 0.83$) and our algorithm is not fully capable of creating this distribution. Thus there is a gap to capacity of approximately $0.4\,\text{dB}$ (see ”shaped TS capacity” in Fig. 3). The polar-coded curve was designed as in [11] and we used an 8 bit CRC, $|\mathcal{D}| = 242$, and SCL encoding and decoding with $L_{\text{SCL}} = 32$. The polar code outperforms the proposed LDPC code (the worse slope could potentially be improved by optimizing the CRC or using a larger list size).

Finally, Fig. 4 shows the FER performance of our encoding scheme with $R = 1/3$ for codes of different length. We see that the proposed scheme works for different lengths (we depict $n_c = 1056$, $n_c = 4224$, and $n_c = 8448$). The fraction of shaping bits compared to the code length was kept constant and for all three cases we obtain the same empirical distribution of the parity bits. For $n_c = 4224$ we further depict a curve for a code with a lower code rate. The obtained distribution is naturally less biased but the code has better error correcting capability and the two curves for $n_c = 4224$ lie very close to each other. This shows that our scheme can, similarly to PAS, also be used for rate adaptivity.

V. Conclusion

We proposed an efficient BP-like algorithm for LLPS using LDPC codes based on the Tanner graph of the systematic generator matrix. It is sub-optimal compared to LLPS with SDM, but still delivers a good FER performance in simulations with standard LDPC codes. The algorithm allows for rate flexibility and compared to other schemes for LDPC codes it always generates a valid codeword. Further work to optimize the algorithm can include tailored code-design and optimizing the decimation strategy and the selection of the shaping bits.

Acknowledgment

The authors thank Gerhard Kramer for helpful comments.

References

[1] R. G. Gallager, *Information theory and reliable communication*. Springer, 1968, vol. 588.

[2] G. D. Forney, “Trellis shaping,” *IEEE Trans. Inf. Theory*, vol. 38, no. 2, pp. 281–300, 1992.

[3] F. Kschischang and S. Pasupathy, “Optimal nonuniform signaling for Gaussian channels,” *IEEE Trans. Inf. Theory*, vol. 39, no. 3, pp. 913–929, 1993.

[4] R. Laroia, N. Farvardin, and S. A. Tretter, “On optimal shaping of multidimensional constellations,” *IEEE Trans. Inf. Theory*, vol. 40, no. 4, pp. 1044–1056, 1994.

[5] D. Xiao, L. Wang, D. Song, and R. D. Wesel, “Finite-support capacity-approaching distributions for AWGN channels,” in *2020 IEEE Inf. Theory Workshop (ITW)*. IEEE, 2021, pp. 1–5.

[6] G. Böcherer, F. Steiner, and P. Schulte, “Bandwidth efficient and rate-matched low-density parity-check coded modulation,” *IEEE Trans. Commun.*, vol. 63, no. 12, pp. 4651–4665, Dec 2015.

[7] G. Böcherer, P. Schulte, and F. Steiner, “Probabilistic shaping and forward error correction for fiber-optic communication systems,” *J. Lightw. Technol.*, vol. 37, no. 2, pp. 230–244, 2019.

[8] P. Schulte and G. Böcherer, “Constant composition distribution matching,” *IEEE Trans. Inf. Theory*, vol. 62, no. 1, pp. 430–434, 2015.

[9] L. Wang, D. Song, F. Areces, T. Wiegart, and R. D. Wesel, “Probabilistic shaping for trellis-coded modulation with CRC-aided list decoding,” *IEEE Trans. Commun.*, vol. 71, no. 3, pp. 1271–1283, 2023.

[10] A. D. Git, B. Matuz, and F. Steiner, “Protograph-based LDPC code design for probabilistic shaping with on-off keying,” in *2019 53rd Annual Conf. Inf. Sciences and Systems (CISS)*, March 2019, pp. 1–6.

[11] T. Wiegart, F. Steiner, P. Schulte, and P. Yuan, “Shaped on–off keying using polar codes,” *IEEE Commun. Letters*, vol. 23, no. 11, pp. 1922–1926, Nov 2019.

[12] J. Honda and H. Yamamoto, “Polar coding without alphabet extension for asymmetric models,” *IEEE Trans. Inf. Theory*, vol. 59, no. 12, pp. 7829–7838, Dec 2013.

[13] M. J. Wainwright and E. Martinian, “Low-density graph codes that are optimal for binning and coding with side information,” *IEEE Trans. Inf. Theory*, vol. 55, no. 3, pp. 1061–1079, March 2009.

[14] M. Mondelli, S. H. Hassani, and R. L. Urbanke, “How to achieve the capacity of asymmetric channels,” *IEEE Trans. Inf. Theory*, vol. 64, no. 5, pp. 3371–3393, May 2018.

[15] V. Aref, N. Macris, and M. Vuffray, “Approaching the rate-distortion limit by spatial coupling with belief propagation and decimation,” in *2013 IEEE Int. Symp. Inf. Theory (ISIT)*, July 2013, pp. 1177–1181.

[16] S. Kumar, A. Vem, K. Narayanan, and H. D. Pfister, “Spatially-coupled codes for side-information problems,” in *2014 IEEE Int. Symp. Inf. Theory (ISIT)*, June 2014, pp. 516–520.

[17] G. Böcherer, D. Lentner, A. Cirino, and F. Steiner, “Probabilistic parity shaping for linear codes,” 2019. [Online]. Available: https://arxiv.org/abs/1902.10648

[18] C. Runge, T. Wiegart, and D. Lentner, “Improved list decoding for polar-coded probabilistic shaping,” in *2023 12th Int. Symp. Topics in Coding (ISTC)*, 2023.

[19] ETSI, “5G NR, multiplexing and channel coding (3GPP TS 38.212 version 16.2.0 release 16),” 2020.

[20] IEEE, “IEEE Standard for Local and Metropolitan Area Networks Part 16,” *IEEE 802.16e*, 2006.

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