Improving the Thresholds of Generalized LDPC Codes With Convolutional Code Constraints

Muhammad Umar Farooq, Alexandre Graell i Amat, Senior Member, IEEE, and Michael Lentmaier, Senior Member, IEEE

Abstract—CC-GLDPC codes are a class of generalized low-density parity-check (GLDPC) codes where the constraint nodes (CNs) represent convolutional codes. This allows for efficient decoding in the trellis with the forward-backward algorithm, and the strength of the component codes easily can be controlled by the encoder memory without changing the graph structure. In this letter, we extend the class of CC-GLDPC codes by introducing different types of irregularity at the CNs and investigating their effect on the BP and MAP decoding thresholds for the binary erasure channel (BEC). For the considered class of codes, an exhaustive grid search is performed to find the BP-optimized and MAP-optimized ensembles and compare their thresholds with the regular ensemble of the same design rate. The results show that irregularity can significantly improve the BP thresholds, whereas the thresholds of the MAP-optimized ensembles are only slightly different from the regular ensembles. Simulation results for the AWGN channel are presented as well and compared to the corresponding thresholds.

Index Terms—LDPC codes, GLDPC codes, convolutional codes, codes on graphs, iterative decoding thresholds.

I. INTRODUCTION

LOW-DENSITY parity-check codes (LDPC) codes are known for their excellent iterative decoding performance at large block lengths. Generalized LDPC (GLDPC) codes are obtained by replacing the weak single parity-check codes at the constraint nodes (CNs) by more powerful component codes. It has been demonstrated in [1] that this can be of advantage in the short block length regime, where LDPC codes suffer from a larger number of short cycles in their Tanner graph. Asymptotically, GLDPC codes based on regular graphs have very good minimum distance growth rates and MAP decoding thresholds, even if the variable node (VN) degree is only two. These properties make them suitable for spatial coupling [2] to benefit from improved BP decoding thresholds due to the threshold saturation effect. Also staircase codes [3], designed for high-speed optical communications, can be viewed as a class of spatially coupled GLDPC codes with higher density and product code structure, similar to braided codes [4]. A class of GLDPC codes with convolutional code (CC) constraints was proposed and analyzed in [5], [6], and [7].

Without spatial coupling, analogously to LDPC codes, the BP decoding thresholds of GLDPC codes can be improved with irregular graphs, either on the VN or the CN side [8]. A simple example of check-hybrid GLDPC codes [9] is the code doping procedure introduced in [10], where some CNs in the protograph of a given LDPC code are replaced by stronger CNs. This procedure is extended in [1] to replacing a tunable fraction of CNs for code optimization. A common challenge in the above mentioned designs of GLDPC codes is that the design rate of the ensembles is changing when CNs are replaced, which makes a fair comparison between codes difficult. Furthermore, it is not straightforward to increase the strength of a component code without changing its rate and length, which in turn would affect the graph structure.

In [11] we introduced a family of regular GLDPC codes with CC constraints that avoids this challenge. These CC-GLDPC codes can be viewed as turbo-like codes with regular graph structure [12] and hence as an extension of braided convolutional codes [13] to arbitrary regular graphs. The aim was to identify the performance trade-off between CC-GLDPC codes with strong component codes and lower VN degree to conventional LDPC codes with larger VN degree.

It was observed that both the minimum distance of CC-GLDPC codes and their MAP thresholds improve with either the strength of the component code trellis or with increasing the VN degree of the graph at the expense of degraded BP thresholds. This type of performance behavior make CC-GLDPC codes suitable for spatial coupling. Indeed, the SC-CC-GLDPC codes were observed numerically to exhibit the threshold saturation phenomenon.

In this letter, we introduce ensembles of irregular CC-GLDPC codes where the CNs represent component codes that may differ in memory, rate and fraction of edges they are connected to. Based on the observations in [11], we intentionally keep the VN degree low but avoid degree-1 VNs which are typical for classical turbo-like codes [12]. More precisely, we use regular degree-two VNs and limit the number of CN types to two.1 Considering the design rates $R = 1/3, 1/2, 2/3$ as example, we perform a density evolution (DE)
An advantage of trellis-based GLDPC codes is that the strength of the component codes can be changed without altering the graph structure, or equivalently the design rate \( R \) of the graph. Under a preserved graph structure, in [11] the strength of the code ensemble was changed by varying the component code strength. Component code trellises with generator polynomials having trellises memories of one, two and three, respectively were used.

An irregular CC-GLDPC w.r.t. the component code trellis strength is obtained when \( m_1 \neq m_2 \), whereas a regular CC-GLDPC w.r.t. the component code trellis strength is obtained when \( m_1 = m_2 \). In this work, we have utilized the trellises with memories \( m_j = 1, 2, 3 \) having generator polynomials \((1, 1/3), (1, 5/7), (1, 13/15)\). Our analysis in Section III reveals that this type of irregularity improves the thresholds only in combination with an irregular distribution of edges and irregular component code rates.

C. Irregular Distribution of Edges

Now consider the case that \( d_c = d_{c,1} = d_{c,2}, N_1 \neq N_2 \), and \( m_1 \neq m_2 \). In this case, both the strengths and the proportions of the component codes \( C_j \) become irregular, but their rates are kept regular. Let us introduce \( \rho_j \), which represents the fraction of the total number of edges that are connected to \( C_j \), which can be expressed as

\[
\rho_j = \frac{d_{c,1} \cdot N_1}{d_{c,1} \cdot N_1 + d_{c,2} \cdot N_2} \quad \rho_2 = 1 - \rho_1 .
\]

A varying \( \rho \) produces a family of codes with a varying proportion of \( C_j \), under a fixed \( R = 1 - 2/d_c \), and fixed component code rates \( r = r_1 = r_2 \).

D. Irregular Component Code Rates

For a given design rate \( R \), an irregularity w.r.t. the component code rates pair \( r = (r_1, r_2) \) is achieved if \( r_1 \neq r_2 \), whereas regularity w.r.t. the rates is achieved if \( r_1 = r_2 \). Regular or irregular component code rates \( r_j \) can be combined with regular or irregular memories \( m_j \), and regular or irregular lifting factors \( N_j \) under the constraint that the design rate \( R \) remains fixed. The design rate in terms of component code rates pair \( r = (r_1, r_2) \), is expressed as

\[
R(\rho, r) = 1 - d_c (1 - r_c(\rho, r)) = 2 r_c(\rho, r) - 1
\]

where \( r_c \) denotes the average component code rate

\[
r_c(\rho, r) = \rho r_1 + (1 - \rho) r_2 .
\]
For a given \( r_1 \) and a fixed design rate \( R \) we can then express \( r_2 \) as a function of \( \rho \), namely
\[
    r_2(\rho) = \frac{r_c - \rho \cdot r_1}{1-\rho} = \frac{(R+1)/2 - \rho \cdot r_1}{1-\rho} \quad . \tag{4}
\]

We remark that both \( r_1 \) and \( r_2 \) are rational fractions, and lie within the interval \([r_m, 1]\), where \( r_m = 1/2 \) is the rate of the mother code trellis used in this work. For a given design rate \( R \) or an averaged component code rate \( r_c \), a family of CC-GLDPC codes with irregular component codes is obtained for \( \rho \in [0, 1] \).

III. Threshold Analysis and Optimization

In this section, we perform a decoding threshold analysis for the BEC and perform an exhaustive search over a discretized parameter space to determine the ensemble parameters resulting in the most competitive BP and MAP thresholds. Such a full search becomes feasible for the BEC, where exact DE equations can be derived. The AWGN channel performance of the BEC-optimized ensembles will be evaluated in Section IV in terms of BER simulations and AWGN channel thresholds.

A. Density Evolution Equations

The mother code transfer functions of the considered generator polynomials can be derived by following the method explained in [12]. Suppose \( f_s \) and \( f_p \) denote the decoder transfer functions of systematic and parity bits of the mother code, then
\[
    p_s^{(i)} = f_s(q_s^{(i)}, q_p^{(i)}), \quad (5) \\
    p_p^{(i)} = f_p(q_s^{(i)}, q_p^{(i)}), \quad (6)
\]
where \( p_s^{(i)} \) and \( p_p^{(i)} \) are the erasure probabilities of the outgoing extrinsic message from the CN, and the \( q_s^{(i)} \) and \( q_p^{(i)} \) are the erasure probabilities of the incoming extrinsic messages to the CN. If the parity bits of the mother code trellis are punctured randomly with a uniform distribution to achieve the desired component code rate, then the incoming erasure probabilities to the transfer functions become
\[
    q_s^{(i)} = q^{(i-1)}, \quad (7) \\
    q_p^{(i)} = \frac{2k_j - n_j}{k_j} \cdot 1 + \frac{n_j - k_j}{k_j} q^{(i-1)}. \quad (8)
\]

After the decoding, the extrinsic probability of the averaged extrinsic message from \( C_j \) is computed using
\[
    p_j^{(i)} = \frac{k_i}{n_j} \cdot p_s^{(i)} + \frac{n_j - k_j}{n_j} \cdot p_p^{(i)}. \quad (9)
\]
Combining the contributions of both decoders we get
\[
    p^{(i)} = \rho_1 p_1^{(i)} + \rho_2 p_2^{(i)} = f_T(q^{(i-1)}), \quad (10)
\]
where the transfer function \( f_T(\cdot) \) follows from applying equations (5)–(9). Finally, including the VN update the DE recursion can be expressed as
\[
    q^{(i)} = \varepsilon \cdot f_T(q^{(i-1)}). \quad (11)
\]
The transfer function of the punctured trellis in (9) is applicable for a rate range of \((n_j)/n_j, \ldots, (n_j - 1)/n_j\).

B. Computation of BP and MAP Thresholds

We use the BP-EXIT function \( h^\text{BP}(\varepsilon) \) [14] to determine the BP and MAP thresholds of CC-GLDPC codes, which based on (11) is expressed in parameterized form as
\[
    h^\text{BP}(\varepsilon) = \left\{ \begin{array}{ll}
        (\varepsilon, 0) & \varepsilon \in [0, \varepsilon^\text{BP}] \\
        (\varepsilon(x), f_T^2(x)) & \varepsilon \in (\varepsilon^\text{BP}, 1) \leftrightarrow x \in (x^\text{BP}, 1)
    \end{array} \right.
\]
since we restrict the VN degree to \( d = 2 \) in this work. Note that \( x^\text{BP} \) corresponds to value of \( x \) for which \( \varepsilon(x) \) is minimum.
The MAP threshold is determined from the BP-EXIT function as
\[
    R(\rho, r) = \int_{\varepsilon^\text{MAP}}^{1} h^\text{BP}(\varepsilon) d\varepsilon.
\]

C. Threshold Optimization

A parameter search space for the design of CC-GLDPC codes is given by the CN parameters: rate pair \( r \), trellis memory pair \( m \), and fraction pair \((\rho, 1-\rho)\). Hereby the possible choices of component code rates and the fraction \( \rho \) for a given design rate \( R \) are interlinked via (3). The restriction to a code graph with two CNs and degree-two VN's results in a symmetry in the component code rates. This symmetry allows us to reduce the search space parameters to \( r_1, r, \) and \( m \). Note that \( r_2 \) is computed from (4) for a given \( \rho \in [0, 1] \) and \( r_1 \in [r_m, 1) \). The graph symmetry in component code rate implies that for a fixed \( \rho \), when \( r_2 \) lies within \([r_m, r_c]\), then the resultant \( r_1 \) lies within \([r_c, 1]\). This holds even when the rate ranges described above are swapped.

 Along the component code rate pair \( r = (r_1, r_2) \) and the fraction pair \((\rho, 1-\rho)\) the parameters are continuous, and to limit the search space we use discretization by considering a lifted code graph with a total number of edges \( n_c \). Sweeping the parameter \( \rho = 2/n_2, \ldots, (n_c - 2)/n_2 \) we then obtain a finite number of possible rate pairs \((r_1, r_2)\) that make an exhaustive grid search possible. The size of \( n_c \) determines the resolution of the component code rates and fractions, and it has to be chosen carefully to balance accuracy and complexity. In our threshold optimization we use values of \( n_c \) close to 512.

D. Results and Discussion

Thresholds of CC-GLDPC codes having regular graph structure \( d_{c,1} = d_{c,2}, N_1 = N_2, \) and both regular and irregular trellis strengths are shown in Table I (left). It can be observed that irregularity w.r.t. trellis strength alone, while keeping the component code degrees and rates regular, does not offer performance improvement compared to the fully regular CC-GLDPC codes.

Figure 3 shows how an irregular distribution of edges, characterized by the parameter \( \rho \), influences the thresholds. Choosing the best parameter \( \rho = \rho^* \) results in the values shown in Table I (right). In this case the BP threshold can be improved compared to the fully regular case.

Now we compute BP and MAP thresholds of rate \( R = 1/3, 1/2, 2/3 \) CC-GLDPC codes over the complete search space, including irregular component code rates. For a given trellis memory pair \( m \), the exhaustive search produces a family of curves \( \varepsilon^\text{BP} \) as a function of \( r_1(\rho) \) for each \( \rho \). The design
Fig. 3. BP and MAP thresholds vs $\rho$ of rate $R = 1/2$ CC-GLDPC codes with irregular trellis strength.

Suppose $\rho^*$ denotes the fraction that produces the best BP threshold in the considered parameter search space. The plot of $\varepsilon^{BP}$ as a function of $r_1(\rho = \rho^*, m)$ for $R = 1/2$ CC-GLDPC is shown in Fig. 4. The plot is obtained by projecting the curves $\varepsilon^{BP}$ of all trellis memory combinations into a single dimension. It can be seen from the figure that $\rho^* = 0.5212$, $m = (3, 2)$, and $r_1 = 0.6554$ constitute the design parameters corresponding to the best BP threshold of rate $R = 1/2$ CC-GLDPC codes. Similarly, we obtain the design configuration with best MAP decoding performance.

The design parameters corresponding to best BP and best MAP thresholds for $R = 1/3, 1/2, 2/3$ are listed in Table II. It is observed that the irregular CC-GLDPC graph structures yield the best BP and MAP decoding thresholds. However, the MAP decoding performance of the regular code ensembles is very similar to that of the MAP-optimized irregular ones, also visible in Table II.

### IV. Finite Length AWGN Channel Performance

We have performed BER simulations for BEC-optimized CC-GLDPC codes of length $n = 102400$ with the ensemble parameters corresponding to the best BEC thresholds w.r.t. the BP and MAP decoding performance listed in Table II. The simulations are compared with AWGN channel thresholds. Table III shows AWGN channel thresholds computed by the erasure channel prediction method used in [15], while the thresholds in Table IV are obtained by Monte Carlo DE based on histograms.

BER simulations for randomly chosen CC-GLDPC codes of design rates $R = 1/3, 1/2, 2/3$ with BP decoding are shown in Fig. 5. The maximum number of decoding iterations was set to 100 in all simulations. Each of the component code trellises was terminated separately, resulting in a small rate loss. Furthermore, the input block lengths were rounded to an integer resulting in a slightly different overall code rate. All codes where constructed with a random permutation over the total number of edges $n_e, 1 + n_e, 2 = 2n$, and also the positions

It should be noted that the BEC-optimized ensembles not necessarily are optimal for the AWGN channel. Our aim is to demonstrate that this approach provides an efficient way of finding ensembles with good AWGN thresholds.
of punctured bits in the mother code trellises were chosen randomly. The AWGN channel thresholds shown in the figure correspond to Table IV.

In general, the BER plots of the BP-optimized codes show a visible error floor and their gaps to the corresponding BP thresholds are larger than for MAP-optimized codes, indicating a slower convergence rate. The error floor is particularly bad for the highest code rate $R = 2/3$. Note that for this ensemble a severe puncturing up to rate $r_2 = 0.9540$ is required. The error floors result from low-weight error patterns in the high-rate component codes, and potentially can be lowered with structured ensembles and a more careful design of the permutations and the puncturing patterns. The BER plots of the MAP-optimized codes, on the other hand, show steep decay in the waterfall regions and a lower gap to the corresponding BP thresholds. For lower code rates, however, they suffer from weak BP thresholds compared to the BP-optimized codes. This gap between the BP thresholds of MAP-optimized and BP-optimized codes gets smaller as the design rate $R$ increases, making MAP-optimized or even regular ensembles more attractive at high rates. At low rates, however, irregularity can significantly improve the performance in the waterfall region.

V. CONCLUDING REMARKS

We have introduced different types of irregularity at the CNs of CC-GLDPC codes in order to improve their BP and MAP decoding thresholds. The proposed ensembles can be flexibly tuned in terms of trellis memory, fraction of edges connected to the different CNs, and the component code rates. An exhaustive grid search was then performed over these three dimensions to optimize the thresholds. Our results show that irregularity can improve the BP thresholds compared to the regular case considered in [11] at the expense of a higher error floor.

The BP thresholds are also better than those of the turbo-like code ensembles considered in [12], and we can avoid the use of degree-1 VNs. At larger rates, it can be observed that the gap between MAP-optimized and BP-optimized ensembles gets small and that irregularity becomes less advantageous.

TABLE IV

| $R$ | Parameter | $E_b/N_0 (\rho^*)$ [dB] |
|-----|-----------|-------------------------|
| 1/3 | $(E_b/N_0)^{BP}$ | 0.0046 | 1.2386 |
| 1/2 | $(E_b/N_0)^{BP}$ | 0.6764 | 1.0616 |
| 2/3 | $(E_b/N_0)^{BP}$ | 1.6037 | 1.5516 |

Fig. 5. BER simulations for $n = 102400$ and $R = 1/3$ (left), $R = 1/2$ (middle), and $R = 2/3$ (right).

REFERENCES

[1] Y. Liu, P. M. Olmos, and D. G. M. Mitchell, “Generalized LDPC codes for ultra reliable low latency communication in 5G and beyond,” IEEE Access, vol. 6, pp. 72002–72014, 2018.

[2] D. G. M. Mitchell, P. M. Olmos, M. Lentmaier, and D. J. Costello, “Spatially coupled generalized LDPC codes: Asymptotic analysis and finite length scaling,” IEEE Trans. Inf. Theory, vol. 67, no. 6, pp. 3708–3723, Jun. 2021.

[3] B. P. Smith, A. Farhood, A. Hunt, F. R. Kschischang, and J. Lodge, “Staircase codes: FEC for 100 Gb/s OTN,” J. Lightw. Technol., vol. 30, no. 1, pp. 110–117, Jan. 1, 2012.

[4] L. M. Zhang, D. Truhachev, and F. R. Kschischang, “Spatially coupled split-component codes with iterative algebraic decoding,” IEEE Trans. Inf. Theory, vol. 64, no. 1, pp. 205–224, Jan. 2018.

[5] I. Andriyanova, J.-P. Tillich, and J.-C. Carlach, “Asymptotically good codes with high iterative decoding performances,” in Proc. IEEE Int. Symp. Inf. Theory (ISIT), Sep. 2005, pp. 850–854.

[6] I. Andriyanova and J.-P. Tillich, “Designing a good low-rate sparse-graph code,” IEEE Trans. Commun., vol. 60, no. 11, pp. 3181–3190, Nov. 2012.

[7] I. Andriyanova, “Finite-length scaling of turbo-like code ensembles on the binary erasure channel,” IEEE J. Sel. Areas Commun., vol. 27, no. 6, pp. 918–927, Aug. 2009.

[8] E. Paolini, M. P. C. Fossorier, and M. Chiani, “Generalized and doubly generalized LDPC codes with random component codes for the binary erasure channel,” IEEE Trans. Inf. Theory, vol. 56, no. 4, pp. 1651–1672, Apr. 2010.

[9] I. P. Mulholland, M. F. Flanagan, and E. Paolini, “Minimum distance distribution of irregular generalized LDPC code ensembles,” in Proc. IEEE Int. Symp. Inf. Theory (ISIT), Jul. 2013, pp. 1670–1674.

[10] G. Liva, W. E. Ryan, and M. Chiani, “Quasi-cyclic generalized LDPC codes with low error floors,” IEEE Trans. Commun., vol. 56, no. 1, pp. 49–57, Jan. 2008.

[11] M. U. Farooq, S. Moloudi, and A. M. Lentmaier, “Generalized LDPC codes with convolutional code constraints,” in Proc. IEEE Int. Symp. Inf. Theory (ISIT), Jun. 2020, pp. 479–484.

[12] S. Moloudi, M. Lentmaier, and A. G. I. Amat, “Spatially coupled turbo-like codes,” IEEE Trans. Inf. Theory, vol. 63, no. 10, pp. 6199–6215, Oct. 2017.

[13] W. Zhang, M. Lentmaier, D. J. Costello, and K. S. Zigangirov, “Braided convolutional codes,” in Proc. Int. Symp. Inf. Theory (ISIT), Adelaide, SA, Australia, Sep. 2005, pp. 592–596.

[14] T. Richardson and R. Urbanke, Modern Coding Theory. New York, NY: USA: Cambridge Univ. Press, 2008.

[15] M. U. Farooq, S. Moloudi, and M. Lentmaier, “Thresholds of braided convolutional codes on the AWGN channel,” in Proc. IEEE Int. Symp. Inf. Theory (ISIT), Jun. 2018, pp. 1375–1379.