Joint Optimization of Multi-Rate LDPC Code Ensembles for the AWGN Channel Based on Shortening and Puncturing

Moritz Beermann and Peter Vary
Institute of Communication Systems and Data Processing (ind), RWTH Aachen University, Germany
{beermann|vary}@ind.rwth-aachen.de

***Abstract*—In most communication systems, being able to adapt the error protection strength of the physical layer is essential to ensure the functionality of the system in potentially strongly varying conditions. Traditional forward error correction code design focuses on the optimization of codes with a fixed code rate for a worst-case channel condition. So-called *rate-compatible* codes, on the other hand, are tailored to support multiple code rates. In this paper we present an optimization strategy for a rate-compatible system based on *shortening* and *puncturing* of Low-Density Parity-Check (LDPC) codes. While the technique itself is well known, we propose a novel joint optimization of the resulting multi-rate LDPC code ensemble at all rates. This joint optimization achieves close to capacity decoding thresholds over an arbitrarily wide range of target code rates using only a single encoder and decoder implementation.**

I. Introduction

In wireless communications the physical layer has to deal with potentially rapidly varying transmission channels. Moreover, the variety of supported services determines the required data rates and error protection levels. The Forward Error Correction (FEC) functionality of such systems commonly allows for an adaptation to these variations in order to efficiently use the available resources as, e.g, the limited radio bandwidth. FEC codes that can achieve different protection levels and information data rates based on one fixed so-called *mother code* are frequently denoted *rate-compatible* (RC).

As one of the most powerful FEC schemes, *Low-Density Parity-Check* (LDPC) codes [1], [2] are included in many of today's communication systems. However, these standards specify separate LDPC codes for each supported code rate. This increases hardware complexity since all supported codes have to be implemented in each transmitter and receiver.

Rather than using multiple codes, there exist techniques to construct RC LDPC codes of higher or lower code rates from a fixed-rate mother code. In [3], Li et al. studied *parity bit puncturing* to construct LDPC codes of higher code rates and proposed a special *code extension* to achieve lower code rates. Another approach to construct codes of lower code rates was presented in [4] using *information shortening*.

In this paper, we present a novel joint optimization strategy for the binary input additive white Gaussian noise (BIAWGN) channel to achieve a multi-rate LDPC code ensemble with one mother code of rate r_m. The resulting multi-rate ensemble supports code rates between 0 and r_m by information shortening and code rates between r_m and 1 by parity puncturing in a rate-compatible way. With the proposed approach, close to capacity performance is achieved over an arbitrarily wide range of code rates.

978-1-4799-3083-8/14/$31.00 © 2014 IEEE

II. Low-Density Parity-Check Codes

A binary sparse parity check matrix $\mathbf{H}$ of dimension $M \times N$ defines a binary (N, K) LDPC code with N code bits and $K = N - M$ information bits. We assume $\mathbf{H}$ to have full rank, such that the code rate is given by $r = \frac{K}{N}$. The *Tanner graph* description of an LDPC code is convenient for describing the commonly used *message-passing* decoders. This graph consists of N variable nodes and M check nodes which are connected by edges according to the non-zero entries of $\mathbf{H}$.

Let the number of non-zero elements in a column (row) of the parity check matrix be denoted the *weight* of that column (row) or equivalently the *degree* of the corresponding variable (check) node. The distribution of the different column and row weights is commonly described by a so-called *degree distribution pair* in the following polynomial notation:

$$\textit{edge perspective: } \lambda(x) = \sum_{i=2}^{d_{v,\max}} \lambda_i x^{i-1}, \quad \rho(x) = \sum_{j=2}^{d_{c,\max}} \rho_j x^{j-1}$$

$$\textit{node perspective: } \Lambda(x) = \sum_{i=2}^{d_{v,\max}} \Lambda_i x^{i}, \qquad R(x) = \sum_{j=2}^{d_{c,\max}} R_j x^{j},$$

where $d_{v,\max}$ is the maximum column weight and $d_{c,\max}$ the maximum row weight. The coefficients Λ_i and R_j of the node perspective correspond to the proportion of columns of weight i and the rows of weight j, respectively. The coefficients λ_i and ρ_j of the edge perspective, on the other hand, describe the proportion of non-zero entries found in columns of weight i and in rows of weight j, respectively. A valid degree distribution has coefficients between zero and one and $\Lambda(1) = R(1) = \lambda(1) = \rho(1) = 1$ holds. The two notations are related by the following two equations:

$$\lambda(x) = \frac{\frac{\mathrm{d}}{\mathrm{d}x}\Lambda(x)}{\frac{\mathrm{d}}{\mathrm{d}x}\Lambda(x)\big|_{x=1}} \quad \text{and} \quad \rho(x) = \frac{\frac{\mathrm{d}}{\mathrm{d}x}R(x)}{\frac{\mathrm{d}}{\mathrm{d}x}R(x)\big|_{x=1}}.$$

The connection between the code rate $r = \frac{K}{N}$ and the degree distribution pair is given by

$$r = 1 - \frac{\sum_{i=2}^{d_{v,\max}} i\Lambda_i}{\sum_{j=2}^{d_{c,\max}} jR_j} = 1 - \frac{\sum_{j=2}^{d_{c,\max}} \rho_j/j}{\sum_{i=2}^{d_{v,\max}} \lambda_i/i}. \quad (1)$$

It is well known that the degree distributions of an LDPC code strongly influence its error correcting capabilities. For long code lengths, close to capacity performance can be achieved by LDPC codes under *Belief Propagation* (BP) decoding if their degree distributions are carefully optimized [5].

III. Rate-Compatible LDPC Codes

In applications that support different code rates, it is desirable to employ *rate-compatible* coding schemes based on one single so-called *mother code*. In this paper, we consider a combination of *information shortening* and *parity puncturing* of a single mother code. This scheme can achieve code rates in an arbitrary range between 0 and 1. It has a very low encoding and decoding complexity, and only a single encoder and decoder implementation are needed.

A similar combination of shortening and puncturing was already used in [4], where the authors successively optimized shortening parameters for a given fixed mother code with a given fixed puncturing scheme. In contrast, the novelty of our approach is the joint optimization of all supported code rates. This includes the optimization of the mother code itself, as well as all parameters of information shortening and parity puncturing. Before this joint rate-compatible optimization process is discussed in Sec. IV, the general concepts of information shortening and parity puncturing are reviewed.

A. Rate-Compatibility by Information Shortening[1]

Information shortening is a technique for achieving lower code rates from a fixed rate mother code.

1) Encoding: The (N, K) mother code of rate r_m takes $K = K_{\text{eff}} + K_d$ bits, composed of K_{eff} information bits and K_d known *dummy bits*, as input and generates $M = \frac{1-r_m}{r_m}(K_{\text{eff}} + K_d)$ parity bits at the output. W.l.o.g. we will assume that all dummy bits are set to 0. Before transmission, the dummy bits are removed from the code word since they are known to the receiver. This results in the shortened $(N - K_d, K - K_d)$ code with effective block length $N_{\text{eff}} = K_{\text{eff}} + M$ and the effective code rate $r_{\text{eff}} = \frac{K_{\text{eff}}}{K_{\text{eff}}+M}$. Any code rate $r_{\text{eff}} \in (0, r_m]$ can be achieved by an accordant choice of $K_{\text{eff}} \in \{1, \dots K\}$ (of course only in discrete steps).

2) Decoding: At the receiver, the dummy bits are known and thus provide perfect *a priori* information that is inserted into the received sequence of log-likelihood ratios (LLRs) in form of the value $+\infty$ (corresponding to the known binary value 0). For the computation of messages according to the check node update of the BP decoder, these messages with infinite reliability do not have to be considered, since $+\infty$ is the neutral element of the *boxplus* operation $\boxplus$ (see, e.g., [7]).

[1]The introductory section III-A follows the corresponding parts of our earlier publication [6].

As there is no need for improving the reliability of a dummy bit's (known) value, also the incoming messages of nodes with perfect knowledge are of no further use. Accordingly, the effective graph as seen by the decoder is simply the mother code graph with all dummy nodes (and all connected edges) removed. Thus, the effective parity check matrix is obtained by removing those columns that are associated to dummy bit positions from the mother code's parity check matrix.

For the asymptotic analysis of a shortened code of rate r_{eff} we refer to the variable node degree distribution of the pruned graph as the *effective* degree distribution (in node perspective)

$$\Lambda^{[r_{\text{eff}}]}(x) = \sum_{i=2}^{d_{v,\max}} \Lambda_i^{[r_{\text{eff}}]} x^i.$$

Given the original degree distribution $\Lambda(x)$ and the effective degree distribution $\Lambda^{[r_{\text{eff}}]}(x)$, the degree distribution of all pruned nodes is given by

$$\begin{aligned}\Delta^{[r_{\text{eff}}]}(x) &= \sum_{i=2}^{d_{v,\max}} \Delta_i^{[r_{\text{eff}}]} x^i \\ &= \sum_{i=2}^{d_{v,\max}} \left(\Lambda_i \frac{1 - r_{\text{eff}}}{r_m - r_{\text{eff}}} - \Lambda_i^{[r_{\text{eff}}]} \frac{1 - r_m}{r_m - r_{\text{eff}}} \right) x^i.\end{aligned}$$

Note that for implementation purpose it is not mandatory to remove any nodes or edges. Existing decoder implementations do not have to be changed and are fully compatible to this rate-matching technique.

B. Rate-Compatibility by Puncturing of Parity Bits

In contrast to information shortening, parity puncturing achieves higher code rates by discarding some of the parity bits generated by a fixed rate mother code prior to transmission.

1) Encoding: The (N, K) mother code of rate r_m takes K input bits and generates $M = \frac{1-r_m}{r_m} K$ parity bits at the output. Then, M_p parity bits are discarded prior to transmission, resulting in an $(N - M_p, K)$ code with effective block length $N_{\text{eff}} = K + M_{\text{eff}}$ and the effective code rate $r_{\text{eff}} = \frac{K}{K+M_{\text{eff}}}$. Any code rate $r_{\text{eff}} \in [r_{\text{eff}}, 1]$ can be achieved by an accordant choice of $M_{\text{eff}} \in \{0, \dots M\}$ (again only in discrete steps).

2) Decoding: At the receiver, no information about the punctured bits is available and thus, zeros are inserted into the received LLR sequence at the punctured positions. LLRs of zero correspond to complete uncertainty as the binary values $0/1$ are equally likely. For the computation of messages according to the BP check node update, these messages with complete uncertainty are dominant in the sense of $L_i \boxplus 0 = 0$ (see, e.g., [7]). Due to this dominance of uncertainty, punctured nodes first have to be *recovered* by neighboring unpunctured nodes before a connected check node can pass non-zero messages. This behavior degrades the decoding performance and has to be considered in the design of efficient puncturing patterns. The theoretical effect of punctured bits was incorporated into the *density evolution* (DE) framework [8] by Ha et al. in [9] and will be used for the joint optimization in Sec. IV.

In accordance with [9], we define the *puncturing polynomial* of the effective rate r_{eff} as

$$\Pi^{[r_{\text{eff}}]}(x) = \sum_{i=2}^{d_{v,\max}} \Pi_i^{[r_{\text{eff}}]} x^i,$$

where $\Pi_i^{[r_{\text{eff}}]}$ denotes the fraction of degree i nodes to be punctured. It should be noted that the coefficients $\Pi_i^{[r_{\text{eff}}]}$ do not sum up to one, but are related to the mother code by

$$\widetilde{\Pi}^{[r_{\text{eff}}]}(x)\Big|_{x=1} = \sum_{i=2}^{d_{v,\max}} \widetilde{\Pi}_i^{[r_{\text{eff}}]} = \sum_{i=2}^{d_{v,\max}} \Pi_i^{[r_{\text{eff}}]} \Lambda_i = 1 - \frac{r_m}{r_{\text{eff}}}. \quad (2)$$

Again, decoder implementations do not have to be modified. LLRs of punctured bits are simply initialized with zeros, before straightforward decoding.

IV. Joint Rate-Compatible Optimization

Given a set of L target code rates $r_1 < r_2 < \cdots r_L$ in an arbitrary range between 0 and 1, the objective of the proposed joint rate-compatible optimization is to find a mother code of rate $r_m \in \{r_1, \ldots r_L\}$ and shortening and puncturing rules to achieve close to capacity performance at all target rates. The output of the optimization consists of L polynomials:

- $\Lambda^{[r_m]}(x) = \Lambda(x)$: a variable node degree distribution for the mother code of rate $r_m \in \{r_1, \ldots r_L\}$,
- $\Lambda^{[r_l]}(x)$: an effective degree distribution for each target code rate $r_l \in \{r_1, \ldots r_{m-1}\}$ with $r_l < r_m$,
- $\Pi^{[r_h]}(x)$: a puncturing polynomial for each target code rate $r_h \in \{r_{m+1}, \ldots r_L\}$ with $r_h > r_m$.

Additionally, at all rates, we assume the effective check node degree distributions to be *concentrated* (i.e., to consist of at most two consecutive degrees). This assumption is valid for puncturing a mother code with concentrated check node degree distribution. For *random shortening*, however, this assumption is not valid. To still achieve near concentrated check node degree distributions, we select the shortened columns during code construction such that the shortened check node distributions are as concentrated as possible.

Furthermore, we reorder the matrix columns by moving all shortened columns to the left and all punctured columns to the right. This allows to always shorten a code word from left to right and puncture parity bits from right to left, as suggested in [4] and depicted in Fig. 1.

A. Cost Function

We assume the performance at each target code rate to be of equal importance. Thus, a candidate solution of L valid polynomials is evaluated by computing the mean square distance of the decoding thresholds (i.e., the lowest channel SNR at which near error free decoding is possible) to the Shannon capacity limit at each of the L rates as cost function.

As described in Sec. III-A2, from the decoder's point of view, a shortened version of the mother code is equivalent to an "unshortened" code with the respective effective degree distribution $\Lambda^{[r_{\text{eff}}]}(x)$. This allows to compute the decoding thresholds of the mother code and all shortened versions by straightforward density evolution (we employed an implementation of *discretized density evolution* [10] using 9 bit quantization). To determine the decoding thresholds of the punctured codes, we adapted the modified density evolution for punctured LDPC codes as proposed for a Gaussian Approximation in [9] to the case of discretized density evolution.

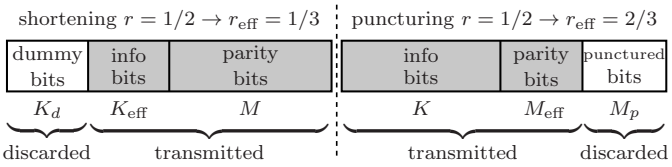

Fig. 1. Exemplary rate $1/2$ code word shortened to $r_{\text{eff}} = 1/3$ (left) and punctured to $r_{\text{eff}} = 2/3$ (right)

By means of the described algorithms, we compute a theoretical decoding threshold t_k for each rate $r_k \in \{r_1, \ldots r_L\}$ of the candidate solution. Let c_k denote the Shannon BIAWGN capacity at code rate r_k. The *gap to capacity* is then simply given by $g_k = t_k - c_k$. Note that all thresholds and gaps are given in dB. Using this notation, the employed cost function C can be written as

$$C\left(\Lambda^{[r_1]}(x), \ldots \Lambda(x), \ldots \Pi^{[r_L]}(x)\right) = \tfrac{1}{L} \textstyle\sum_{k=1}^{L} g_k^2.$$

Using the squared distance ensures a certain "flatness" of a solution's sequence of gaps g_k, since an improvement at positions with large gaps is considered more important than at positions with smaller gaps. By optimizing all rates jointly, it is possible to accept a degradation at one rate to enable an improvement at other rates, which might be necessary due to the constraints between the different rates (see next section).

Unequal importance levels of target code rates (e.g., due to more or less preferred operating points) can be incorporated into the optimization by assigning an appropriate weight w_k to each code rate yielding $C'(\cdot) = \frac{1}{L} \sum_{k=1}^{L} w_k g_k^2$.

B. Constraints

In the following, the five constraints 1) – 5) that are necessary for the joint optimization to yield a valid rate-compatible solution are explained. The interrelation of the involved quantities are visualized in Fig. 2.

1) Since shortening is performed from left to right, the shortened columns of each rate must be included in the ones of the next lower rate (for all degrees i):

$$\frac{\Lambda_i}{1 - r_m} \geq \frac{\Lambda_i^{[r_{m-1}]}}{1 - r_{m-1}} \geq \cdots \geq \frac{\Lambda_i^{[r_1]}}{1 - r_1}.$$

This is equivalent to $\Delta_i^{[r_{\text{eff}}]} \geq 0$.

2) When shortening from one rate r_k to the next lower one r_{k-1} $\left(\text{for } k \in \{2, \ldots m\}\right)$, the number of shortened nodes of any degree i cannot exceed the total number of shortened nodes for that specific rate difference:

$$(1 - r_{k-1})\, \Lambda_i^{[r_k]} - (1 - r_k)\, \Lambda_i^{[r_{k-1}]} \leq r_k - r_{k-1}.$$

This is equivalent to $\Delta_i^{[r_{\text{eff}}]} \leq 1$.

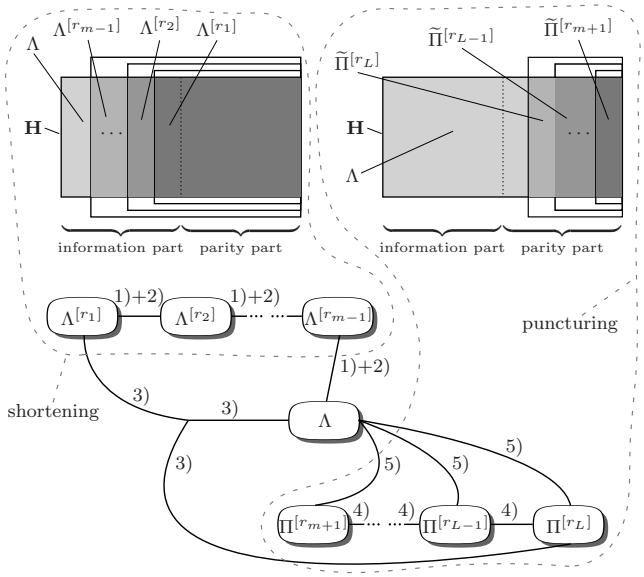

Fig. 2. Visualization of relevant submatrices and rate-compatible constraints

3) Punctured and shortened nodes must be disjunct and, thus, the number of punctured nodes of each degree i at the highest rate r_L must be less or equal to the number of degree i nodes remaining in the degree distribution of the lowest shortened rate r_1:

$$\Pi_i^{[r_L]}\Lambda_i \leq \frac{1-r_m}{1-r_1}\Lambda_i^{[r_1]}$$

4) Since puncturing is performed from right to left, the punctured nodes of each rate must be included in the ones of the next higher rate (for all degrees i):

$$\Pi_i^{[r_{m+1}]} \leq \Pi^{[r_{m+2}]} \leq \cdots \leq \Pi_i^{[r_L]}.$$

5) For each rate $r_h \in \{r_{m+1}, \dots r_L\}$ the puncturing polynomial $\Pi^{[r_h]}(x)$ in conjunction with the mother code degree distribution $\Lambda(x)$ must result in the correct total number of punctured bits:

$$\widetilde{\Pi}^{[r_h]}(1) = \sum_{i=2}^{d_{v,\max}} \Pi_i^{[r_h]}\Lambda_i = 1 - \frac{r_m}{r_h}.$$

A summary of the optimization problem is given in Fig. 3. It includes the trivial constraints of all coefficients lying between 0 and 1 and the degree distributions' coefficients summing up to 1 (cf. constraint 6) in Fig. 3). Furthermore, we also apply the well known *stability condition* of Richardson et al. [5] (cf. constraint 7) in Fig. 3). As stated earlier, all check node degree distributions are assumed to be concentrated to two consecutive degrees, which allows to compute them given the target code rate, the variable node degree distribution, and (1).

C. Optimization Process

Many previously proposed optimization strategies formulate the problem of finding an optimal degree distribution as *linear program* (e.g., [5], [11]) which can be solved efficiently with existing problem solvers. To achieve a linear program formulation, these strategies are usually based on certain assumptions

$$\frac{1}{L}\sum_{k=1}^{L}(t_k - c_k)^2 \rightarrow \min$$

s.t.
1) $\dfrac{\Lambda_i^{[r_l]}}{1-r_l} \geq \dfrac{\Lambda_i^{[r_{l-1}]}}{1-r_{l-1}} \quad \forall\, l \in \{2, \dots m\}$
2) $(1-r_{l-1})\,\Lambda_i^{[r_l]} - (1-r_l)\,\Lambda_i^{[r_{l-1}]} \leq r_l - r_{l-1} \quad \forall\, l \in \{2, \dots m\}$
3) $\Pi_i^{[r_L]}\Lambda_i \leq \dfrac{1-r_m}{1-r_1}\Lambda_i^{[r_1]}$
4) $\Pi_i^{[r_h]} \leq \Pi^{[r_{h+1}]} \quad \forall\, h \in \{m+1, \dots L-1\}$
5) $\displaystyle\sum_{i=2}^{d_{v,\max}} \Pi_i^{[r_h]}\Lambda_i = 1 - \frac{r_m}{r_h} \quad \forall\, h \in \{m+1, \dots L\}$
6) $\Lambda^{[r_l]}(1) = 1,\ \Lambda_i^{[r_l]} \geq 0 \quad \forall\, l \in \{1, \dots m\}$
7) $\lambda_2^{[r_l]} < \dfrac{e^{\frac{1}{2\sigma_n^2}}}{\sum_j \rho_j^{[r_l]}(j\text{–}1)}, \sigma_n^2 = \dfrac{10^{\frac{-t_l}{10}}}{2r_l}\ \forall\, l \in \{1, \dots m\}$
8) t_l is the lowest decodable SNR determined by density evolution for $\Lambda^{[r_l]}(x),\ \forall\, l \in \{1, \dots m\}$
9) t_h is the lowest decodable SNR determined by modified density evolution for $\Lambda(x)$ punctured with $\Pi^{[r_h]}(x),\ \forall\, h \in \{m+1, \dots L\}$

Fig. 3. Joint rate-compatible optimization problem

about the probability density functions of the messages that are passed during decoding. However, these assumptions are not valid in general. Still, it has been observed that for medium to high code rates, these optimization strategies yield very good solutions. However, for low code rates (e.g., lower than 0.5), the assumptions made by these approaches are considerably violated and lead to large errors in the predicted achievable decoding thresholds [11]. Since the multi-rate optimization proposed in this paper is designed to work well in the whole range of code rates between 0 and 1, we set aside these kinds of approximations. Instead we resort to discretized density evolution for determining the decoding thresholds and use a *Simulated Annealing* (SA) [12] strategy to find a solution of the optimization problem. While SA is only a heuristic that does not necessarily find the global optimum, very good results can be achieved for the given problem, though. To reduce the optimization time, splitting the SA process into two phases as depicted in Fig. 4 has proven useful:

- Left: Degree distributions $\Lambda^{[r_l]}$ ($l \in \{1, \dots m\}$) and puncturing distributions $\Pi^{[r_h]}$ ($h \in \{m+1, \dots L\}$) are jointly optimized at all rates subject to all constraints in Fig. 3. The current solution vector consiting of all distributions is randomly modified in each step, and the cost function C is evaluated. If an improvement was found, the modified vector will always be accepted as new current solution, while a degradation by ΔC is only accepted with probability $P = \exp(-\Delta C/(c \cdot T))$, where T denotes the current "temperature" of the SA process

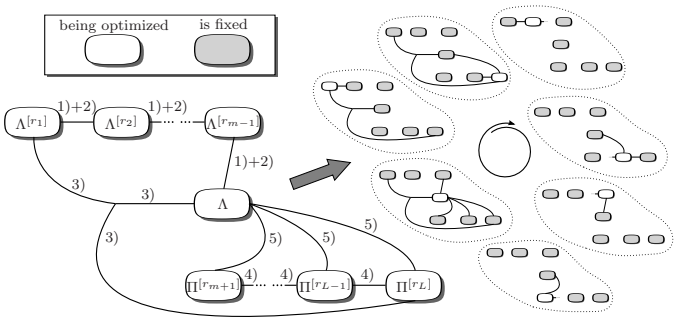

Fig. 4. Active constraints of two-phase *Simulated Annealing* process.

and c an empirically determined constant.

- Right: Starting at rate r_k that has the largest current gap to capacity g_k and ending at the rate with the smallest current gap to capacity, the optimization process tries to improve each rate without violating any constraints in an iterative round-robin scheme.

The first phase tries to find the "deepest" local minimum by slowly cooling down the temperature T to avoid getting stuck in local minima. In the second phase, only downhill movements are allowed to make sure the final solution is maneuvered towards the bottom of the current local minimum.

V. Results

To confirm the effectiveness of the proposed optimization strategy, we present results for the challenging case of covering the range of (relevant) code rates r_{eff} between 0.1 and 0.9 in steps of 0.1 using only a single mother code and a maximum variable node degree of $d_{v,\max} = 10$. For comparison, we employ two different previous results. Firstly, the rate-compatible scheme presented in [4] with the same set of code rates, the same maximum degree, and a mother code rate of 0.5. Secondly, the results of [11, Fig. 10] as a benchmark as to the best achievable performance for $d_{v,\max} = 10$ if dedicated degree distributions are optimized for each of the target code rates without any rate-compatible features.

Using the proposed scheme, we have optimized multi-rate ensembles based on mother codes with rates between 0.2 and 0.8. Fig. 5 exemplarily shows the performance of five setups:

- **M2** (blue +): Proposed optimization with $r_m = 0.2$
- **M5** (red □): Proposed optimization with $r_m = 0.5$
- **M8** (green ∗): Proposed optimization with $r_m = 0.8$
- **Tian** (black): Reference results from [4] with r_m = 0.5
- **Chung** (dotted): achievable gaps for independent codes without rate-compatibility from [11, Fig. 10].

Detailed optimization results of setups *M5* and *M8* are given below in Tab. I. In the upper plot of Fig. 5, the gaps $g_k = t_k - c_k$ of the density evolution thresholds t_k to the corresponding channel capacity c_k (in dB) as produced by the optimization algorithm are depicted for all effective code rates r_{eff}. It can be seen that for the four rate-compatible setups, the respective minimal gap is achieved at the mother code rate and the gap increases when moving away from this rate. The values C (see IV-A) of the mean square gap are

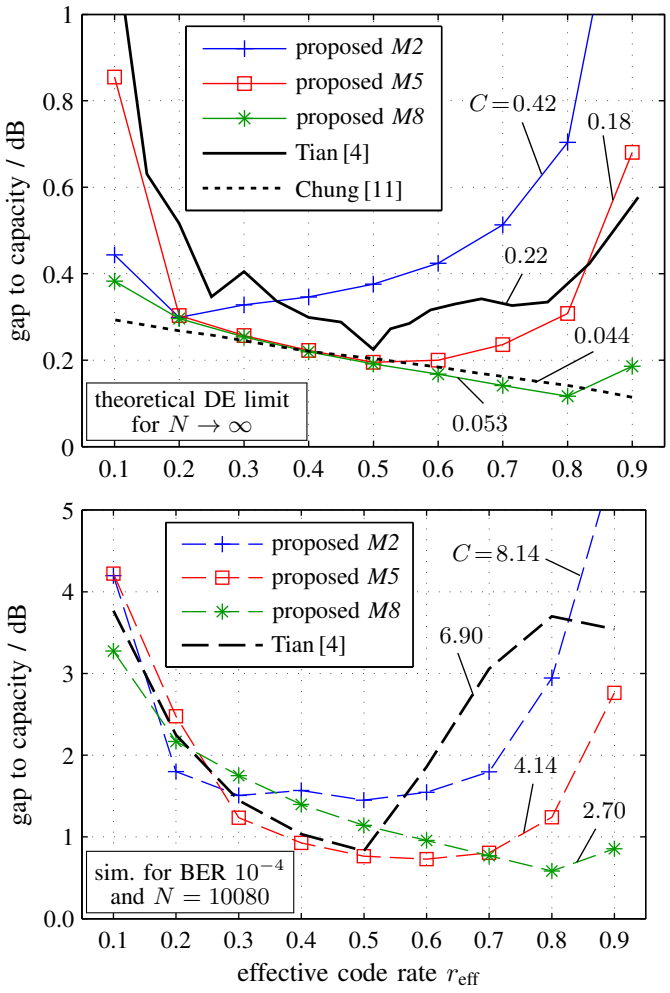

Fig. 5. Top: gaps of density evolution (DE) thresholds to BIAWGN capacity over code rate. Bottom: gaps of E_b/N_0 values for bit error rate 10^{-4} to BIAWGN capacity with 100 BP iterations over code rate. Each curve is labeled with its mean square value according to cost function C in IV-A.

given for all five curves. The proposed setup *M5* outperforms the reference results *Tian* with the same mother code rate in terms of the mean square gap and also at each single effective code rate except 0.9. Setup *M2* seems to strongly suffer from the fact, that a large range of code rates from 0.3 to 0.9 has to be achieved by puncturing. The best performance of the rate-compatible schemes is achieved by setup *M8* with a mean square gap of only 0.053 which is very close to the benchmark of 0.044 for the scheme *Chung* which does not offer any rate-compatibility. As stated earlier, these results confirm that the proposed optimization strategy yields close to optimal performance since it comes very close to the case of unconstrained individual code optimization.

In addition to the DE results, we have simulated bit error rates (BER) for each of the rate-compatible setups by using the *Progressive Edge Growth* (PEG) algorithm [13] to construct a parity check matrix of code length $N = 10080$. The input degree sequence of the PEG algorithm was sorted in such a way that the optimized degree distributions and puncturing fractions are achieved by shortening bits from left to right and

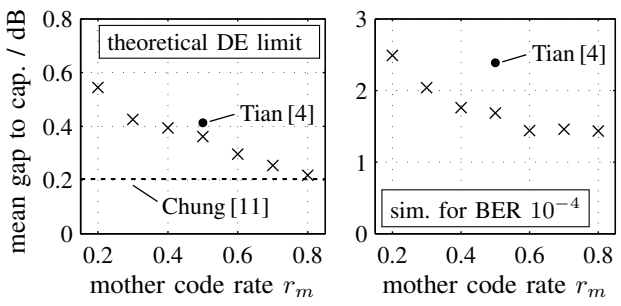

Fig. 6. Mean gap to capacity achieved by the five investigated setups with different mother code rates r_m. DE thresholds (left) and BER 10^{-4} (right).

puncturing bits from right to left as shown in Fig. 1. This way, we also ensure the check node degree distributions of the shortened codes to be approximately check concentrated. In the lower plot of Fig. 5, again the gaps $g_k^{\text{sim}} = t_k^{\text{sim}} - c_k$ to the channel capacity are shown for the four setups. Here, t_k^{sim} denotes the E_b/N_0 ratio for a BIAWGN channel at which a BER of 10^{-4} is achieved by a *belief propagation* (BP) decoder with 100 iterations. It can be seen that the shape of the curves roughly matches the shape of the corresponding DE results. Again, setup *M8* shows the best performance in terms of the mean square gap with a value of only 2.7.

In Fig. 6 the mean gap to capacity for code rates $0.1, 0.2, \ldots 0.9$ is shown as achieved by the proposed optimization algorithm when different mother code rates are used. Again, the gaps are shown for the DE thresholds (left) and the E_b/N_0 values at which BER 10^{-4} is achieved (right). These results clearly suggest that the proposed optimization strategy favors mother code rates close to the upper limit of the supported effective code rate range, i.e., the resulting shortened codes seem to be more effective than the punctured ones. This observation is confirmed by the BER simulation results, which is especially noteworthy since the average effective code length N_{eff} becomes smaller when the used mother code rate moves closer to the boundaries of the code rate range. For $r_m = 0.8$ and $N = 10080$, e.g., the shortened code of effective rate $r_{\text{eff}} = 0.1$ only has a code length of $N_{\text{eff}} = 2240$. Finally, the fully rate-compatible proposed scheme was shown not only to outperform setup *Tian* but even to approach the performance of the unconstrained optimization of individual codes according to *Chung* if a high rate mother code is used.

VI. Conclusion

We have presented a novel joint optimization algorithm that yields a multi-rate LDPC code ensemble supporting an arbitrarily wide range of code rates, requiring only a single encoder/decoder hardware implementation. The approach is based on jointly optimizing the degree distribution of a mother code, the effective degree distributions of shortened versions of this code, and the puncturing fractions to be applied to achieve higher code rates. We have shown that by using a two-phase Simulated Annealing process, we are able to find such multi-rate ensembles that have close to capacity performance over the whole range of relevant code rates and outperform previously proposed rate-compatible schemes that provide the same flexibility. The presented results suggest that by choosing the rate of the mother code close to the upper limit of the code rate regime, the performance of a system using independent dedicated codes for each rate can on average be approached to within about 0.01 dB in terms of DE thresholds.

TABLE I
Optimized multi-rate ensemble for $r_m = 0.5$ and $r_m = 0.8$

	degree	2	3	4	5	10	t_k/dB	g_k/dB
$r_m = 0.5$	$\Lambda^{[0.1]}$	0.669	0.199	0.0	0.053	0.08	-0.43	0.86
	$\Lambda^{[0.2]}$	0.625	0.242	0.001	0.049	0.083	-0.66	0.30
	$\Lambda^{[0.3]}$	0.571	0.270	0.005	0.044	0.110	-0.36	0.26
	$\Lambda^{[0.4]}$	0.529	0.297	0.004	0.039	0.131	-0.02	0.22
	Λ	0.490	0.320	0.010	0.035	0.145	0.38	0.19
	$\Pi^{[0.6]}$	0.205	0.084	0.0	0.292	0.202	0.88	0.20
	$\Pi^{[0.7]}$	0.349	0.176	0.0	0.486	0.285	1.50	0.24
	$\Pi^{[0.8]}$	0.435	0.307	0.0	0.595	0.297	2.35	0.31
	$\Pi^{[0.9]}$	0.531	0.346	0.0	0.824	0.306	3.88	0.68
$r_m = 0.8$	$\Lambda^{[0.1]}$	0.687	0.218	0.029	0.001	0.064	-0.90	0.38
	$\Lambda^{[0.2]}$	0.620	0.235	0.051	0.002	0.092	-0.67	0.30
	$\Lambda^{[0.3]}$	0.572	0.257	0.045	0.011	0.116	-0.36	0.25
	$\Lambda^{[0.4]}$	0.532	0.279	0.041	0.014	0.133	-0.02	0.22
	$\Lambda^{[0.5]}$	0.498	0.297	0.040	0.015	0.150	0.38	0.19
	$\Lambda^{[0.6]}$	0.460	0.328	0.033	0.012	0.167	0.85	0.17
	$\Lambda^{[0.7]}$	0.426	0.355	0.026	0.009	0.184	1.41	0.14
	Λ	0.390	0.386	0.017	0.006	0.200	2.16	0.12
	$\Pi^{[0.9]}$	0.136	0.098	0.343	0.034	0.071	3.38	0.19

References

[1] R. G. Gallager, *Low-Density Parity-Check Codes*. Cambridge, MA, USA: M.I.T. Press, 1963.

[2] D. J. C. MacKay and R. M. Neal, "Good Codes Based on Very Sparse Matrices," *Cryptography and Coding, 5th IMA Conference*, Springer, Berlin, Germany, 1995, pp. 100–111.

[3] J. Li and K. Narayanan, "Rate-Compatible Low Density Parity Check Codes for Capacity-Approaching ARQ Schemes in Packet Data Communications," *Proceedings Int. Conf. on Communications, Internet, and Information Technology*, Virgin Islands, USA, Nov. 2002, pp. 201–206.

[4] T. Tian and C. R. Jones, "Construction of Rate-Compatible LDPC Codes Utilizing Information Shortening and Parity Puncturing," *EURASIP J. Wirel. Commun. Netw.*, vol. 5, pp. 789–795, Oct. 2005.

[5] T. Richardson, M. Shokrollahi, and R. Urbanke, "Design of Capacity-Approaching Irregular Low-Density Parity-Check Codes," *IEEE Trans. Inform. Theory*, vol. 47, no. 2, pp. 619–637, Feb. 2001.

[6] M. Beermann and P. Vary, "Breaking Cycles with Dummy Bits: Improved Rate-Compatible LDPC Codes with Short Block Lengths," *Proceedings of International ITG Conference on Systems, Communications and Coding*, München, Germany, Jan. 2013.

[7] J. Hagenauer, E. Offer, and L. Papke, "Iterative Decoding of Binary Block and Convolutional Codes," *IEEE Trans. Comm.*, vol. 42, no. 2, pp. 429–445, Mar. 1996.

[8] T. Richardson and R. Urbanke, "The Capacity of Low-Density Parity-Check Codes Under Message-Passing Decoding," *IEEE Trans. Inform. Theory*, vol. 47, no. 2, pp. 599–618, Feb. 2001.

[9] J. Ha, J. Kim, and S. McLaughlin, "Rate-Compatible Puncturing of Low-Density Parity-Check Codes," *IEEE Trans. Inform. Theory*, vol. 50, no. 11, pp. 2824–2836, Nov. 2004.

[10] S. Chung, G. D. Forney, T. J. Richardson, and R. Urbanke, "On the Design of Low-Density Parity-Check Codes within 0.0045 dB of the Shannon Limit," *IEEE Comm. Lett.*, vol. 5, no. 1, pp. 58–60, Feb. 2001.

[11] S. Chung, T. J. Richardson, and R. Urbanke, "Analysis of Sum-Product Decoding of Low-Density Parity-Check Codes Using a Gaussian Approximation," *IEEE Trans. Inform. Theory*, vol. 47, no. 2, pp. 657–670, Feb. 2001.

[12] S. Kirkpatrick, C. Gelatt Jr., and M. Vecchi, "Optimization by Simulated Annealing," *Science*, vol. 220, no. 4598, pp. 671–680, May 1983.

[13] X. Y. Hu, E. Eleftheriou, and D. M. Arnold, "Regular and Irregular Progressive Edge-Growth Tanner Graphs," *IEEE Trans. Inform. Theory*, vol. 51, no. 1, pp. 386–398, Jan. 2005.