Protograph-Based Raptor-Like LDPC Codes for the Binary Erasure Channel

Kasra Vakilinia
Department of Electrical Engineering
University of California, Los Angeles
Los Angeles, California 90024
Email: vakiliniak@ucla.edu

Dariush Divsalar
Jet Propulsion Laboratory
California Institute of Technology
Pasadena, California 91109
Email: Dariush.Divsalar@jpl.nasa.gov

Richard D. Wesel
Department of Electrical Engineering
University of California, Los Angeles
Los Angeles California 90024
Email: wesel@ee.ucla.edu

***Abstract*—This paper designs protograph-based Raptor-like (PBRL) codes as a class of rate-compatible (RC) LDPC codes for binary-erasure channels (BEC). Similar to the Raptor Codes, the RC property is achieved by X-OR operations of the precoded bits. The additional parity bits, which lower the rate, are selected such that their connections in the protograph optimize the density evolution threshold. In order to avoid problematic graphical objects in the CPEG lifted bipartite graph and guarantee the linear growth distance property some constraints are imposed in the threshold optimization algorithm. Simulation results are presented for information block sizes of $k = 1032$, and $k = 16384$. These results are compared with finite blocklength bounds of Polyanskiy, Poor, Verdu (PPV) as well as several asymptotic bounds. The $k = 1032$ code family operates at various rates in the range of $8/9$ to $8/48$ and has an average normalized threshold gap of 5.56% from capacity. The $k = 16384$ code family operates at rates $8/10$ to $8/32$ and has an average normalized threshold gap of 3.27% from capacity.**

I. Introduction

THIS paper provides a general technique for constructing families of rate-compatible (RC) low-density parity-check (LDPC) codes and provides numerical analysis and simulation results to show their outstanding performance.

Low-Density Parity-Check (LDPC) codes were first introduced by Gallager in his dissertation in 1963 [1]. These codes are identified by their parity-check matrices. Irregular LDPC codes have parity-check matrices with various number of ones (weights) in each column. By optimizing the variable-nodes' column weights and check-nodes' row weights (degree distribution), Luby et al. [2] showed that irregular LDPC codes can achieve rates very close to capacity. Richardson et al. [3] invented an algorithm called density evolution (DE) to design and analyze the optimal degree distribution of infinitely long LDPC codes. There are many papers analyzing LDPC codes over BEC. For examples see [4], [5], [6], [7], [8], [9], and [10].

Traditionally, rate compatible (RC) codes such as RC punctured convolutional (RCPC) and RC punctured turbo (RCPT) codes are designed with optimized puncturing patterns which result in good error rate performance across the family of rates. In general, any RC code may be considered as a low-rate code that is punctured to produce higher rates. However, in this paper we use the method of *extension* where the higher-rate codes are designed first and then the lower-rate codes are designed based on their higher-rate counterparts. We use density-evolution/EXIT threshold maximization as the main design criterion to select the connections between variable and check nodes.

As in [11], [12], our RC code family is designed by extending a protograph. After using density evolution to properly design the protograph, a copy-and-permute operation, often referred to as "lifting", is applied to obtain larger graphs of various sizes, resulting in longer-blocklength LDPC codes. Refer to [13], [14] for a thorough discussion on photographs and lifting algorithms. Similar to [15], [16], this paper restricts the code family to have the basic structure of Raptor codes [17]. Constraining the design in this way makes the construction and optimization manageable while still providing outstanding performance and extensive rate-compatibility.

This paper considers a class of RC LDPC codes called protograph-based raptor-like (PBRL) LDPC codes that were introduced in [15] and [16]. The construction and optimization of PBRL codes are discussed and simulation results are presented. The PBRL codes show outstanding performance while providing extensive rate-compatibility.

In [16] the PBRL approach is applied to the design of RC LDPC families over AWGN channels. Also, reciprocal channel approximation (RCA) for AWGN channel replaces the standard density evolution of [15] to provide a fast and accurate approximation of the density evolution threshold to speed up the optimization process. In both [15] and [16] the circulant progressive edge growth (CPEG) algorithm [18] was used for lifting.

This paper extends the design of raptor-like codes over binary input AWGN channels [15] and [16] to the binary erasure channel (BEC). The design of new code families over the BEC for $k = 1032$ and $k = 16384$ code families use CPEG lifting. The performance of these codes is compared with finite-length performance bounds and approximations. Furthermore, we discuss how constraints on the connections

This work is supported by the National Science Foundation under Grant Number 1162501 and by NSF grant CCF-1161822 (JPL Task Plan 82-17473). Any opinions, findings, conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation. This research was carried out in part at the Jet Propulsion Laboratory, California Institute of Technology, under a contract with NASA.

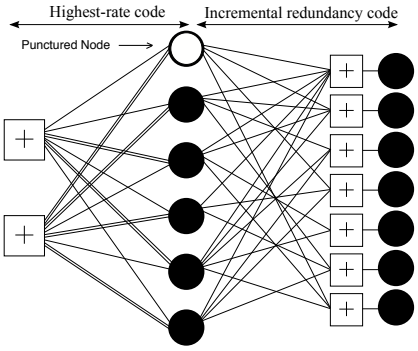

Fig. 1. Protograph for a PBRL code with a punctured node and a highest-rate code (HRC) with rate $4/5$ followed by an incremental redundancy code (IRC) that uses only degree-one variable nodes. The IRC provides lower rates as more of its variable nodes are included, starting from the top.

is used to sacrifice some threshold performance to avoid problematic error floors.

The rest of the paper is organized as follows: Sec. II presents the PBRL code structure. Sec. III provides the design procedure to construct PBRL codes. Sec. IV provides a summary of existing tight bounds for finite length regime. Sec. V constructs examples of PBRL code families and presents analysis and simulation results. Finally Sec. VI concludes the paper.

II. Protograph-Based Raptor-Like LDPC Code

This section introduces the structure, encoding, and decoding of PBRL codes. The parity check matrix of a protograph is called protomatrix which can be presented by a bipartite graph. Let $\mathbf{0}$ be the all-zeros matrix and $\mathbf{I}$ be the identity matrix, the protomatrix of PBRL codes has a general form of

$$H = \begin{bmatrix} H_{\text{HRC}} & \mathbf{0} \\ H_{\text{IRC}} & \mathbf{I} \end{bmatrix}, \tag{1}$$

where HRC describes the highest-rate protograph and IRC corresponds to the incremental redundancy protograph.

Fig. 1 shows the protograph structure of a PBRL code with HRC part on the left and IRC part on the right. The IRC part provides lower rates as gradually more of its variable nodes are transmitted, starting from the first variable node on the top. The use of the punctured node as shown in the protogaph of Fig. 1 improves the iterative decoding threshold.

The protomatrix of the protograph shown in Fig. 1 has HRC and IRC parts of

$$H_{\text{HRC}} = \begin{bmatrix} 1 & 1 & 2 & 1 & 2 & 1 \\ 2 & 2 & 1 & 2 & 1 & 2 \end{bmatrix} \tag{2}$$

and

$$H_{\text{IRC}} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 \end{bmatrix}. \tag{3}$$

The first column of the above matrices corresponds to the punctured node. The final protograph determined by HRC and IRC protographs is lifted to produce the actual code. The CPEG algorithm uses circulant matrices in the lifting process while guaranteeing a minimum girth. After lifting, the HRC code is structurally similar to the precode in a Raptor code.

The IRC is created by the addition of variable nodes that are exclusively degree-one, which correspond to the identity matrix in (1). As discussed in [19], there are two main concerns of low minimum distance and high error floors associated with using only degree-1 variable nodes in IRC. Regarding the low minimum distance concern, due to range-equivalency, the set of valid codewords is not affected by restricting the IRC to be diagonal instead of being lower triangular. The other concern that degree-one variable nodes might introduce high error floors is relieved since IRC is concentrated with HRC. As observed in [20], [21] the error floor problem associated with degree-1 variable nodes can be resolved by concatenation with another code.

The Raptor-like structure is quite restrictive. However, these structural constraints do not limit the performance of PBRL codes compared to less-restrictive RC LDPC codes obtained by extension. One of the main conclusions of our paper is that we obtain Raptor-like protographs with very good iterative decoding thresholds despite these constraints. Similar observations for AWGN channels have been made in [19] where the resulting finite-length PBRL codes outperform existing RC LDPC codes that have been designed without the constraint of a Raptor-like structure.

The PBRL code family always transmits the output symbols of the HRC and has deterministic connections in the IRC. These two properties facilitate joint decoding of the HRC and IRC parts. This idea first appeared in [22] for Raptor codes. In order to significantly reduce complexity, for high-rate PBRL codes, the decoder can deactivate those check nodes in the IRC part for which the neighboring degree-one variable node is not transmitted.

The simulation results of Sec. V are obtained by using iterative BP decoding. A more efficient decoder for BEC is the peeling decoder. If the number of iterations in BP decoder is high enough, the performance of these two decoding methods is the same. For BEC, it is also possible to use maximum likelihood decoding at the expense of higher complexity.

Encoding of PBRL codes is as efficient as that of Raptor codes: after encoding the HRC, the encoding of the IRC part only involves XOR operations on the precode output symbols. For efficient encoding of the precode, see the discussion in [14] on efficient encoding of protograph codes.

III. Optimization of PBRL LDPC Codes

This section presents optimization procedures for finding the HRC and IRC components that comprise a good PBRL code family for the BEC. The optimization criteria for both long and short blocklengths is primarily based on maximizing the iterative decoding threshold (channel erasure probability) over

the BEC at each rate while enforcing constraints on the connections to avoid problematic error floors. These constraints are more stringent for short blocklength designs than for long blocklength designs.

For the iterative decoding threshold computations, we use the reciprocal channel approximation (RCA) algorithm. After presenting the RCA in subsection III-A, subsection III-D describes the design of the HRC and subsection III-E describes the design of the IRC.

A. Density Evolution with Reciprocal Channel Approximation

The asymptotic *iterative decoding threshold* [23] characterizes the performance of the ensemble of LDPC codes with the same protograph. This threshold for BEC indicates the maximum channel erasure probability p_{it} to have the expected bit error rate go to zero as the blocklength grows to infinity.

The RCA (reciprocal-channel approximation) algorithm, originally proposed in [24] for regular LDPC codes, is a fast and accurate alternative to density evolution. RCA uses of a single parameter to approximately characterize the distribution of messages exchanged between variable and check nodes over a channel such as AWGN or BEC.

B. Density Evolution in Protographs for the Erasure Channel

For the erasure channel a single real-valued parameter, the probability of erasure p, serves as a stand-in for full density evolution. Alternatively, we can use the RCA algorithm and track the self-information of an erasure, $s = -\ln p$, which is additive at variable nodes. The reciprocal parameter, $r = -\ln(1-p)$, the self-information of a non-erasure, is additive at the check nodes. The parameter r satisfies $C(s)+C(r) = 1$, which implies $e^{-s} + e^{-r} = 1$ where C is the capacity of the channel. We note that $r = R(s)$ and $s = R(r)$ are related to each other by the self-inverting function $R(s) = C^{-1}(1 - C(s)) = -\ln(1 - e^{-s})$ where C^{-1} is inverse of C. With the above definitions for the BEC, the RCA represents the exact one-dimensional density evolution.

To apply density evolution to a protograph we first identify all transmitted variable nodes and select a target channel erasure probability $p_{ch} = e^{-s_{ch}}$. As shown in Fig. 2 messages $\vec{s}_e$ are passed along edges leaving variable nodes ($\vec{s}_e = s_{ch}$ from transmitted nodes and $\vec{s}_e = 0$ from punctured nodes). The transformation $R(\vec{s}_e)$ is applied and an extrinsic return message, $\overleftarrow{r}_e$, is determined by computing the sum of all incoming messages save the one along edge e. Transformation $R(\cdot)$ is then reapplied to produce $\overleftarrow{s}_e$. The process continues and the iterative decoding threshold $p_{it} = e^{-s_{ch}}$ is determined by the smallest value of s_{ch} for which unbounded growth of all messages $\vec{s}_e$ can be achieved.

Algorithm 1 (RCA): Let e_v (e_c) be the set of edges connected to the variable node v (check node c). For iterations $n = 0, \ldots, N$, and for all edges e in the protograph, the RCA computes the messages as follows:

0) For edges e connected to punctured variable nodes, set $\vec{s}_e^{(0)} = 0$. For all other edges set $\vec{s}_e^{(0)} = s_{\text{ch}} = -\ln(p_{ch})$.
1) For $n = 1, \ldots, N$, compute the following:

$$\vec{r}_e^{(n-1)} = R\left(\vec{s}_{e'}^{(n-1)}\right) \tag{4}$$

$$\overleftarrow{r}_e^{(n)} = \sum_{e' \in e_c \setminus e} \vec{r}_{e'}^{(n-1)}, \tag{5}$$

$$\overleftarrow{s}_e^{(n)} = R\left(\overleftarrow{r}_{e'}^{(n)}\right) \tag{6}$$

$$\vec{s}_e^{(n)} = \vec{s}_e^{(0)} + \sum_{e' \in e_v \setminus e} \overleftarrow{s}_{e'}^{(n)}, \tag{7}$$

2) When $n = N$ compute $S_v^{(N)}$ for all variable nodes v

$$S_v^{(N)} = \sum_{e \in e_v} \overleftarrow{s}_e^{(N)}, \tag{8}$$

For large enough N, and some large T, the maximum p_{ch} that produces $S_v^{(N)} > T$ for all variable nodes represents the iterative decoding threshold p_{it} for BEC.

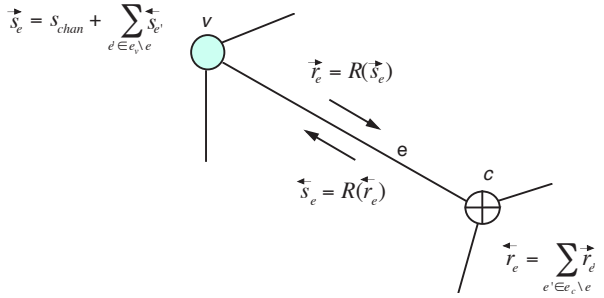

Fig. 2. Density evolution in protographs for the BEC.

C. Maximum a posteriori probability (MAP) threshold of Protographs for the Erasure Channel

To compute an upper bound to the maximum a posteriori probability (MAP) threshold we use the area theorem [25]. To do so we use the results in the final step of RCA algorithm described in the previous subsection for binary erasure channel by computing $S_v^{(N)}$ for each transmitted variable node for range of $p_{ch} \in [p_{it}, 1]$. Starting with $p_{ch} = 1$ and decreasing it, for each value of p_{ch}, we compute $h^{BP}(p_{ch}) = \frac{1}{|v|-|v_p|} \sum_{v' \in v \setminus v_p} e^{-S_{v'}^{(N)}}$ where the $h^{BP}(p_{ch})$ is the output extrinsic erasure probability when the input channel erasure probability is p_{ch}, where v is the set of all variable nodes in the photograph and v_p is the set of punctured variable nodes. Using the area theorem we can obtain the solution to ([26], [10], [9], [4])

$$R_c = \int_{p^*_{MAP}}^{1} h^{BP}(p_{ch}) dp_{ch} \tag{9}$$

Then the p^*_{MAP} is an upper bound to the MAP threshold p_{MAP}. For some ensembles (e.g. regular LDPC), $h^{BP}(p_{ch}) = h^{MAP}(p_{ch})$ for $p_{MAP} \leq p_{ch} \leq 1$ in such case $p^*_{MAP} = p_{MAP}$ ([26], [10]). Let $p_{cap} = 1 - R_c$, where R_c is the code

rate, then $p_{it} \leq p_{MAP} \leq p_{cap}$.

D. Optimizing the Highest-Rate Code of a PBRL Family

Primarily, the HRC photograph is simply the protograph of a good code at the desired rate. Our design follows the work of Divsalar et al. [14, Sec.III], with some additional optimization through RCA analysis and LDPC code simulation.

The main conclusions of [14] are that protograph ensembles with a minimum variable-node degree of 3 or higher are guaranteed to have linear minimum distance growth with the blocklength. Furthermore, as explained in [14], puncturing a node in the HRC can improve the threshold performance. The HRC protographs designed in Secs. V-A and V-B ensure the linear minimum distance growth property and have a punctured node.

E. Optimizing the IRC Protograph of a PBRL Family

The IRC part is optimized by using the RCA algorithm of Sec. III-B and choosing the new rows that optimize the iterative thresholds. The other considerations that are taken into account are the linear minimum distance growth and low error-floor properties enforced by the additional constraints.

PBRL code families retain linear minimum distance growth by requiring that each new row in H_{IRC} has at least two nonzero values. As long as the HRC has linear minimum distance growth, ensuring non-trivial connections for each new check node in the IRC in this way preserves linear minimum distance growth for all rates.

There are two key features that dramatically improve protograph thresholds at low rates. The first is that parallel edges improve the threshold and the second is that the punctured node of the precode should connect to all (or almost all) of check nodes in the IRC part with at least a single edge.

IV. Upper and Lower Bounds for BEC

Gallager [1] proposed an upper bound on the probability of error for discrete memoryless channels. The upper bound can be expressed in terms of Kullback-Leibler distance. The relative entropy of p^* with respect to p, also called the Kullback-Leibler distance, is defined by

$$D(p^*, p) \triangleq p^* \ln \frac{p^*}{p} + (1-p^*) \ln \frac{1-p^*}{1-p}. \tag{10}$$

If $p^* = p_{cap} = 1 - R_c$, which is maximum erasure probability at capacity of BEC, and $p < p^*$ is the channel erasure probability, the FER for a random (n, k) code with code rate $R_c = \frac{k}{n}$ is given by

$$P_e \leq e^{-nD(p_{cap}, p)}. \tag{11}$$

The PPV [27] achievable upper bound on the probability of error is

$$\begin{aligned} \bar{P_e} \quad \leq \quad & 1 - \sum_{j=0}^{n} \binom{n}{j} (1-p)^j p^{n-j} \sum_{m=0}^{2^{nR}-1} \frac{1}{m+1} \binom{2^{nR-1}}{m} \\ & \times 2^{-jm} (1-2^{-j})^{2^{nR}-1-m}. \end{aligned} \tag{12}$$

The PPV [27] lower bound using *meta converse* is given by

$$P_e \geq \sum_{e=[n-k]+1}^{n} \binom{n}{e} (1-p)^{n-e} p^e (1-2^{n-k-e}). \tag{13}$$

The following theorem is also due to PPV [27].

Theorem 2: For the BEC with erasure probability p,

$$nR_c = n(1-p) - \sqrt{np(1-p)} Q^{-1}(\epsilon) + O(1), \tag{14}$$

or $\epsilon \approx Q((1-p-R_c)\sqrt{\frac{n}{p(1-p)}})$, regardless of whether ϵ is maximal or average probability of error.

The above PPV approximation is used to compare with the simulation results for $k = 1032$ and $k = 16384$ raptor-like codes for various code rates in Fig. 4 and Fig. 5.

V. Design Examples

This section presents two families of RC PBRL codes. Subsection V-A presents the design procedure for a short-blocklength codes ($k = 1032$) and subsection V-B presents the design procedure for a long blocklength code ($k = 16384$).

A. Short-Blocklength PBRL Design Example

This subsection provides an example PBRL protograph family designed for information blocklength $k = 1032$. The HRC uses the protograph

$$H_{\text{HRC}}^{(1032)} = \begin{bmatrix} 3 & 3 & 3 & 1 & 1 & 1 & 3 & 1 & 2 & 1 \\ 1 & 1 & 1 & 3 & 3 & 3 & 1 & 2 & 1 & 2 \end{bmatrix}. \tag{15}$$

The IRC adds up to 39 degree-one variable nodes to provide a range of rates of the form $8/(9+i)$ from $8/9$ to $8/48$. The first variable node in HRC part is always punctured. The H_{IRC} obtained from optimizing the threshold is given by (16).

Consistent with [14], [19], the degrees of the variable nodes in HRC protograph are either three or four. Chen et al. in [19] address the trade-off between the threshold performance and error floor over BIAWGN channel by varying the number of degree-4 nodes in the HRC photograph and conclude that high error floors are unavoidable with fewer than 6 degree-4 variable nodes.

After the design of the HRC protomatrix, IRC protomatrix is obtained by optimizing the threshold for each successive rate in a greedy fashion under the constraints on edges to ensure low error floor. For the protomatrices of (15) and (16), Table I shows p_{it}, p^*_{MAP}, and p_{cap}. p_{it} is the highest channel erasure probability that an iterative BP decoder can support as the blocklength of the LDPC code grows to infinity. p^*_{MAP} is the highest channel erasure probability that an MAP decoder supports asymptotically. $p_{cap} = 1 - R_c$ is the maximum erasure probability at capacity of BEC. The "Gap" column of Table I shows the gap between the iterative threshold and the capacity in normalized number of bits. For $k = 1032$ code family the average gap is 0.0556.

Fig. 3 shows an example of EXIT function using RCA to compute the iterative decoding threshold p_{it}. p^*_{MAP} is calculated using the area theorem as an upper bound on MAP threshold. The shaded region under the iterative BP curve has an area equal to the code rate R_c.

$$H_{\text{IRC}}^{(1032)} = \left[\begin{array}{cccccccccc}
2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\
2 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 \\
1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 \\
2 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 \\
1 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 \\
2 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\
1 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\
1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\
1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\
1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 \\
1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\
1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\
1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\
1 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\
1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\
1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\
1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\
1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\
1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\
1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1
\end{array}\right] \tag{16}$$

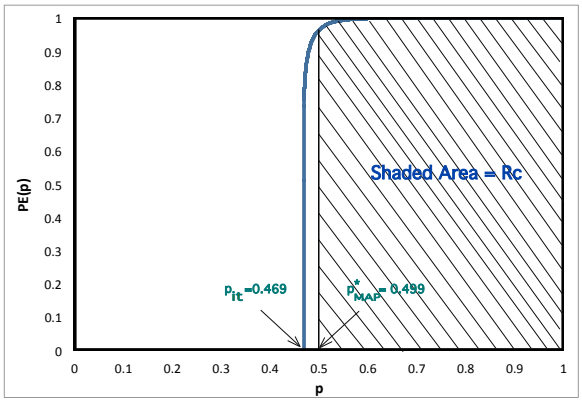

Fig. 3. EXIT function using RCA for the rate-1/2 code of (15) and (16).

After obtaining the HRC and IRC parts, two-step CPEG lifting as in [14] is used to lift the lowest rate code with a required girth of six for the $k = 1032$ family. The pre-lifting step has a lifting number of 3 to remove the parallel edges. The lifting number in the second stage is 43, resulting in an information blocklength of $k = 1032$.

TABLE I
THRESHOLDS OF THE PBRL CODE FAMILY FOR K=1032

Rate	Threshold p_{it}	Threshold p^*_{MAP}	Capacity p_{cap}	Gap $\frac{(1-p_{it}-R_c)}{(1-p_{it})}$
8/9	0.069	0.111	0.111	0.045
8/10	0.167	0.200	0.200	0.040
8/11	0.223	0.264	0.273	0.064
8/12	0.287	0.327	0.333	0.065
8/13	0.343	0.380	0.385	0.064
8/14	0.401	0.427	0.429	0.047
8/15	0.437	0.465	0.467	0.053
8/16	0.469	0.499	0.500	0.058
8/17	0.503	0.528	0.529	0.053
8/18	0.524	0.554	0.556	0.066
8/19	0.556	0.578	0.579	0.052
8/20	0.577	0.598	0.600	0.054
8/21	0.592	0.617	0.619	0.066
8/22	0.617	0.635	0.636	0.050
8/23	0.634	0.651	0.652	0.051
8/24	0.648	0.665	0.667	0.052
8/25	0.663	0.678	0.680	0.051
8/26	0.676	0.691	0.692	0.051
8/27	0.688	0.702	0.704	0.051
8/28	0.699	0.713	0.714	0.051
8/29	0.709	0.723	0.724	0.052
8/30	0.719	0.732	0.733	0.052
8/31	0.728	0.741	0.742	0.052
8/32	0.736	0.749	0.750	0.053
8/33	0.744	0.757	0.758	0.054
8/34	0.751	0.764	0.765	0.054
8/35	0.758	0.771	0.771	0.055
8/36	0.764	0.777	0.778	0.057
8/37	0.770	0.783	0.784	0.058
8/38	0.776	0.789	0.790	0.059
8/39	0.782	0.794	0.795	0.060
8/40	0.787	0.799	0.800	0.061
8/41	0.792	0.804	0.805	0.063
8/42	0.797	0.809	0.810	0.063
8/43	0.802	0.813	0.814	0.060
8/44	0.807	0.817	0.818	0.058
8/45	0.811	0.821	0.822	0.059
8/46	0.815	0.825	0.826	0.060
8/47	0.819	0.829	0.830	0.060
8/48	0.823	0.832	0.833	0.058

Fig. 4 shows the FER performance of the PBRL codes constructed in this section for $k = 1032$. These simulations use floating-point iterative decoders with a flooding schedule for message-passing. Decoding terminates early if all parity checks are satisfied before reaching the maximum number (300) of iterations (as mentioned above, if the number of iterations in BP decoding is large enough then its performance approaches the peeling decoder for BEC). It is surprising that a greedy threshold optimization algorithm alone can produce a family with an outstanding performance.

B. A Long-Blocklength PBRL Design Example

This subsection designs a PBRL code family with block-length $k = 16384$. The protograph family provides a range of rates of the form $8/(10 + i)$ from $8/10$ to $8/32$. The HRC

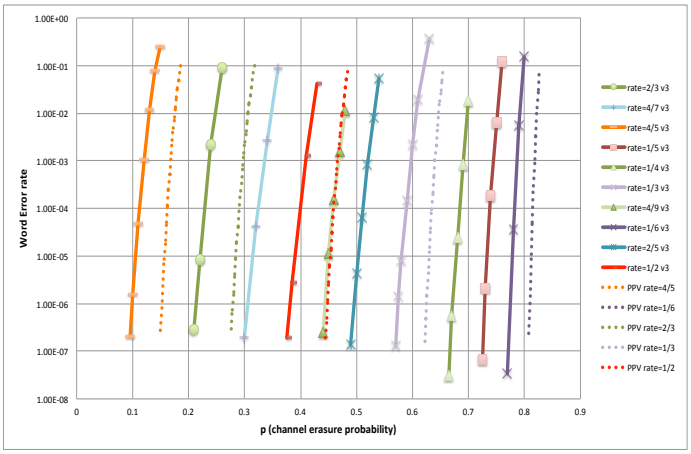

Fig. 4. Frame error rates of PBRL codes with $k = 1032$. The protograph for the PBRL code is based on (15) and (16). The PPV approximation is also shown for several rates.

protograph for the $k = 16384$ PBRL code is

$$H_{\text{HRC}}^{(16384)} = \begin{bmatrix} 3 & 2 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 1 \\ 1 & 1 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 1 & 1 \\ 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 2 & 0 \end{bmatrix}. \quad (17)$$

Similar to design of $k = 1032$ code family, RCA algorithm optimizes the threshold corresponding to the connections of the IRC check nodes to the variable nodes of the HRC part of (18). In the optimization process each connection to the punctured node may have zero, one, or two edges leading to the 0, 1, and 2 values in the first column of (18). Each connection to a non-punctured node may have zero, or one edge leading to the 0, and 1, values in the other columns of (18). Table II shows p_{it}, p^*_{MAP}, and p_{cap} for the $k = 16384$ code family with an average gap of 0.0327 across various rates.

Two step CPEG lifting which guarantees a girth of eight

$$H_{\text{IRC}}^{(16384)} = \begin{bmatrix} 2 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 2 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 2 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ 2 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 2 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 2 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 2 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 2 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (18)$$

TABLE II
THRESHOLDS OF THE PBRL CODE FAMILY

Rate	Threshold p_{it}	Threshold p^*_{MAP}	Capacity p_{cap}	Gap $\frac{(1-p_{it}-R_c)}{(1-p_{it})}$
8/10	0.167	0.200	0.200	0.040
8/11	0.256	0.273	0.273	0.022
8/12	0.319	0.332	0.333	0.021
8/13	0.370	0.384	0.385	0.023
8/14	0.414	0.428	0.429	0.025
8/15	0.452	0.466	0.467	0.027
8/16	0.487	0.499	0.500	0.025
8/17	0.517	0.529	0.529	0.026
8/18	0.542	0.555	0.556	0.030
8/19	0.565	0.578	0.579	0.032
8/20	0.587	0.599	0.600	0.031
8/21	0.606	0.618	0.619	0.033
8/22	0.623	0.636	0.636	0.035
8/23	0.638	0.651	0.652	0.039
8/24	0.655	0.666	0.667	0.034
8/25	0.668	0.679	0.680	0.036
8/26	0.681	0.691	0.692	0.035
8/27	0.692	0.703	0.704	0.038
8/28	0.703	0.713	0.714	0.038
8/29	0.713	0.723	0.724	0.039
8/30	0.722	0.732	0.733	0.041
8/31	0.731	0.741	0.742	0.041
8/32	0.739	0.749	0.750	0.042

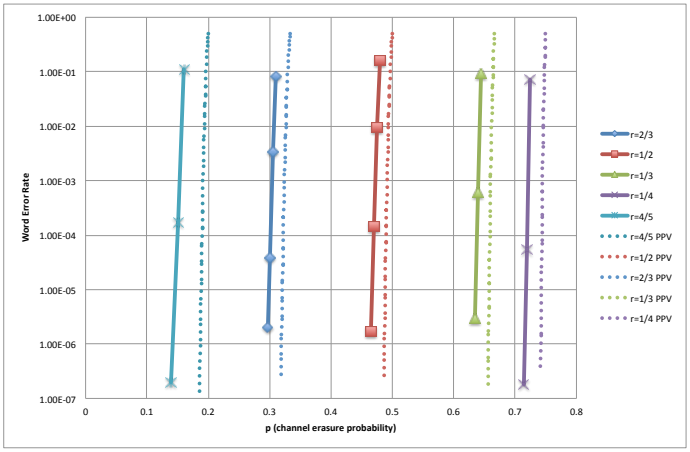

Fig. 5. Frame error rates for the protographs defined by (17)-(18). The lifting number is 2048, resulting in $k = 16384$.

is used to lift the lowest rate code. The pre-lifting step has a lifting number of 4 to remove the parallel edges. The lifting number in the second stage is 512, giving information blocklength of $k = 16384$.

Fig. 5 shows the FER performance of the PBRL codes constructed in this section for $k = 16384$. This figure also shows the finite blocklength analysis of PPV [27].

Fig. 6 illustrates the η values for each rate in Table II. A pair (d_{ACE}, η) implies that every cycle consisting of up to d_{ACE} variable nodes, i.e. every cycle of length up to $2d_{\text{ACE}}$, is connected to at least η extrinsic check nodes. For ACE algorithm see [28]. For example, all the cycles with length 10 or less in $k = 1032$ code have at least 6 extrinsic connections for the rate-1/2 code. Similarly, all the cycles with length 12

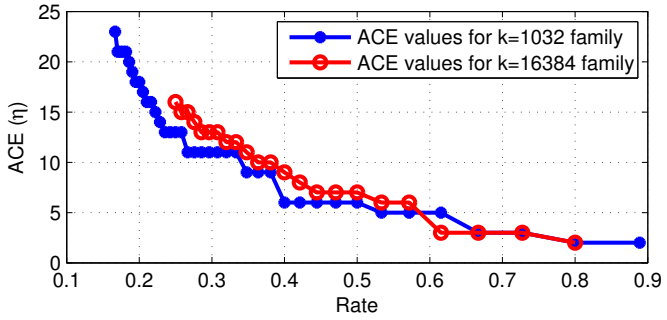

Fig. 6. ACE value (η) versus rate for the long blocklength $k = 16384$ code.

or less in $k = 16384$ code have at least 7 extrinsic connections for the rate-1/2 code.

VI. Concluding Remarks

This paper studies the construction and optimization of protograph-based Raptor-like (PBRL) LDPC codes over the BEC. This paper designs PBRL codes by designing a highest-rate code (HRC) and sequentially adding degree-one variable nodes whose neighboring check node is connected to the variable nodes of the HRC so as to maximize the density evolution threshold (erasure probability). The new connections must obey constraints that control the error floor. Puncturing a single variable node in the HRC improves the threshold performance of PBRL codes.

Instead of the original density evolution, the RCA algorithm (which is exact for BEC) is used to obtain a fast and accurate thresholds of PBRL codes to result in reasonable code-design complexity.

Controlling the error floor is especially important for short-blocklength PBRL code families. For an example short-blocklength PBRL code family designed for $k = 1032$ we sacrificed threshold in order to improve error floor performance by increasing the number of degree-4 variable nodes in the HRC.

In summary, this paper provides a complete design procedure for constructing rate-compatible LDPC code families that perform uniformly close to the capacity for the finite blocklength performance limits for both short ($k = 1032$) and long ($k = 16384$) blocklengths.

References

[1] R. G. Gallager, "Low-density parity-check codes," Ph.D. dissertation, MIT, Cambridge, MA, 1963.

[2] M. Luby, M. Mitzenmacher, A. Shokrollahi, and D. Spielman, "Improved low-density parity-check codes using irregular graphs," *IEEE Trans. Inform. Theory*, vol. 47, pp. 585–598, Feb. 2001.

[3] T. J. Richardson, M. A. Shokrollahi, and R. L. Urbanke, "Design of capacity-approaching irregular low-density parity-check codes," *IEEE Trans. Inf. Theory*, vol. 47, no. 2, pp. 618–637, Feb 2001.

[4] G. Liva, B. Matuz, E. Paolini, and M. Chiani, "Achieving a near-optimum erasure correction performance with low-complexity ldpc codes," *Int. J. Satell. Commun. Network.*, vol. 28, no. 3-4, pp. 236–256, Oct. 2010.

[5] E. Paolini, G. Liva, B. Matuz, and M. Chiani, "Maximum likelihood erasure decoding of ldpc codes: Pivoting algorithms and code design," *IEEE Trans. Commun.*, vol. 60, no. 11, pp. 3209–3220, Nov. 2012.

[6] H. Pfister and I. Sason, "Accumulate-repeat-accumulate codes: Capacity-achieving ensembles of systematic codes for the erasure channel with bounded complexity," *Information Theory, IEEE Transactions on*, vol. 53, no. 6, pp. 2088–2115, June 2007.

[7] M. Lentmaier, M. Tavares, and G. Fettweis, "Exact erasure channel density evolution for protograph-based generalized ldpc codes," in *Information Theory, 2009. ISIT 2009. IEEE International Symposium on*, June 2009, pp. 566–570.

[8] H. Pishro-Nik and F. Fekri, "On decoding of low-density parity-check codes over the binary erasure channel," *Information Theory, IEEE Transactions on*, vol. 50, no. 3, pp. 439–454, March 2004.

[9] C. Measson, A. Montanari, and R. Urbanke, "Maxwell construction: The hidden bridge between iterative and maximum a posteriori decoding," *Information Theory, IEEE Transactions on*, vol. 54, no. 12, pp. 5277–5307, Dec 2008.

[10] C. W. Wang and H. Pfister, "Upper bounds on the map threshold of iterative decoding systems with erasure noise," in *Turbo Codes and Related Topics, 2008 5th International Symposium on*, Sept 2008, pp. 7–12.

[11] S. Dolinar, "A rate-compatible family of protograph-based LDPC codes built by expurgation and lengthening," in *Proc. International Symposium on Information Theory*, 2005, pp. 1627–1631.

[12] T. Nguyen, A. Nosratinia, and D. Divsalar, "The design of rate-compatible protograph LDPC codes," *IEEE Trans. Commun.*, vol. 60, no. 10, pp. 2841–2850, 2012.

[13] J. Thorpe, "Low Density Parity Check (LDPC) codes constructed from protographs," *JPL IPN Progress Report*, vol. 42–154, Aug. 2003.

[14] D. Divsalar, S. Dolinar, C. R. Jones, and K. Andrews, "Capacity-approaching protograph codes," *IEEE J. Sel. Areas Commun.*, vol. 27, No. 6, pp. 876–888, Aug. 2009.

[15] T.-Y. Chen, D. Divsalar, J. Wang, and R. D. Wesel, "Protograph-based raptor-like LDPC codes for rate-compatibility with short blocklengths," in *Proc. IEEE Global Communications Conference*, Houston, TX, Dec. 2011.

[16] T.-Y. Chen, D. Divsalar, and R. D. Wesel, "Protograph-based raptor-like LDPC codes with low thresholds," in *Proc. IEEE International Conference of Communications*, Ottawa, Canada, Jun. 2012.

[17] A. Shokrollahi, "Raptor codes," *IEEE Trans. Inf. Theory*, vol. 52, no. 6, pp. 2551–2567, 2006.

[18] Z. Li and B. V. Kumar, "A class of good quasi-cyclic low-density parity check codes based on progressive edge growth graph," in *Signals, Systems and Computers, 2004. Conference Record of the Thirty-Eighth Asilomar Conference on*, vol. 2. IEEE, 2004, pp. 1990–1994.

[19] T.-Y. Chen, K. Vakilinia, D. Divsalar, and R. D. Wesel, "Protograph-based raptor-like ldpc codes," *CoRR*, vol. abs/1403.2111, 2014. [Online]. Available: http://arxiv.org/abs/1403.2111

[20] J. Garcia-Frias and W. Zhong, "Approaching shannon performance by iterative decoding of linear codes with low-density generator matrix," *IEEE Commun. Lett.*, vol. 7, no. 6, pp. 266–268, June 2003.

[21] A. Eckford, J. Chu, and R. Adve, "Low-complexity cooperative coding for sensor networks using rateless and LDGM codes," in *IEEE International Conference on Communications, 2006. ICC '06.*, vol. 4, June 2006, pp. 1537–1542.

[22] A. Venkiah, C. Poulliat, and D. Declercq, "Analysis and design of raptor codes for joint decoding using information content evolution," in *Proc. International Symposium on Information Theory (ISIT)*, Jun. 2007, pp. 421–425.

[23] T. J. Richardson and R. L. Urbanke, "The capacity of low-density parity-check codes under message-passing decoding," *IEEE Trans. Inf. Theory*, vol. 47, no. 2, pp. 599–618, Feb. 2001.

[24] S. Y. Chung, "On the construction of some capacity-approaching coding schemes," Ph.D. dissertation, MIT, Cambridge, MA, 2000.

[25] A. Ashikhmin, G. Kramer, and S. ten Brink, "Extrinsic information transfer functions: model and erasure channel properties," *Information Theory, IEEE Transactions on*, vol. 50, no. 11, pp. 2657–2673, Nov 2004.

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

[27] Y. Polyanskiy, H. Poor, and S. Verdú, "Channel coding rate in the finite blocklength regime," *Information Theory, IEEE Transactions on*, vol. 56, no. 5, pp. 2307 –2359, May. 2010.

[28] T. Tian, C. Jones, J. Villasenor, and R. Wesel, "Selective avoidance of cycles in irregular ldpc code construction," *Communications, IEEE Transactions on*, vol. 52, no. 8, pp. 1242–1247, Aug 2004.