Design of rate-compatible irregular LDPC codes based on edge growth and parity splitting

Noah Jacobsen and Robert Soni
Alcatel-Lucent
Whippany, NJ 07981
E-mail: {jacobsen,rsoni}@alcatel-lucent.com

***Abstract*— This paper considers the design of rate-compatible low-density parity-check (LDPC) codes with optimized degree distributions for their corresponding rates. The proposed design technique is based on extension, where a high-rate base code, or daughter code, is progressively extended to lower and lower rates such that each extension code is compatible with the previously obtained codes. Specifically, two well-known parity matrix construction methodologies, edge growth and parity splitting, are adapted to yield a flexible framework for constructing rate-compatible parity check matrices with a uniform performance characteristic. The design examples provided are based on extrinsic information transfer (EXIT) chart optimizations and demonstrate good performance up to rates as low as $1/5$.**

I. Introduction

Irregular LDPC codes [1], [2], [3], and related constructions [4], [5], [6], are known to exhibit better performance with respect to turbo-code benchmarks on a variety of channels, especially independent and identically distributed (i.i.d.) channel models. Such codes, based on a sparse and random parity structure, are able to address wide ranges of information block sizes and rates, while being amenable to efficient hardware implementation. For these reasons, LDPC codes are likely to become much more prevalent in forthcoming wireless communication systems.

Families of rate-compatible error-correcting codes are useful in different settings in communications engineering. For example, the hybrid-ARQ protocol is employed to combat fading in cellular systems. Due to channel variability, it is often more efficient to require multiple fast re-transmissions, as provided by hybrid-ARQ, to ensure a successful decoding, rather then provisioning for worst case channel conditions. In the case of wireless vehicular technologies, fading rates are extremely dynamic, and so rate-compatible codes are particularly well-suited.

The conventional approach for obtaining a family of rate-compatible codes is to start with a mother code of low-rate and to selectively puncture redundant bits in order to obtain various codewords of differing length. The problem with such an approach is two-fold: (1) the mother code is typically optimized for efficient operation at low-rates and subsequently exhibits a widening gap to capacity as the amount of puncturing increases, and (2) optimizations of code structure and puncturing patterns are treated separately which is suboptimal. These shortcomings are addressed with the proposed design technique as follows: Starting from a high-rate base code, referred to as the "daughter code," rate-compatible parity check matrices are obtained progressively by extension, in order of decreasing rate, so as to implicitly solve the problem of puncturing. Further, the degree distribution of any given sub-code is optimized for its corresponding rate.

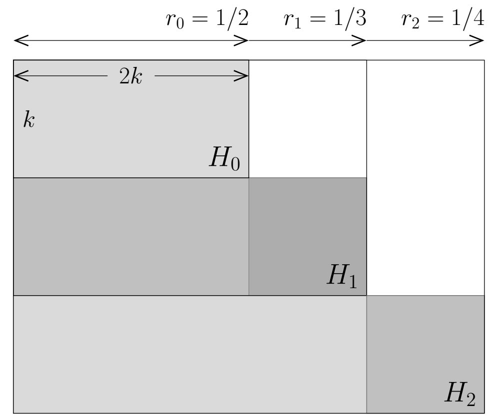

Fig. 1. Rate-compatible parity matrices by extension

Rate-compatible parity matrices produced by extension, illustrated in Figure 1, may alternatively be viewed as punctureless codes, since any given transmission can be decoded with its corresponding parity sub-matrix, rather than by inserting zero LLR-values into a decoder operating on the largest parity matrix (of lowest rate). The proposed design technique, a hybrid of edge growth [7] and parity splitting [8], demonstrates that the extension framework is capable of producing rate-compatible LDPC codes with a uniform gap to capacity over a wide range of rates.

Prior work on this topic includes [9] where puncturing is proven to work well for constructing compatible families of LDPC codes over a limited range of rates, especially for high code rates, but suffers as the range is expanded. In [10], a hybrid puncturing/extension approach is proposed but does not exhibit the desired dynamic range or uniform performance characteristic. Finally, the design in [11] proposes puncturing in combination with information shortening for achieving code rates less than one-half.

II. Design technique

The goal of this design is to produce compatible parity-check matrices of flexible and dynamic rate. The proposed technique extends from a daughter code parity matrix using a hybrid of constrained edge growth and parity splitting, as described in detail in this section. The resulting rate-compatible LDPC codes exhibit a uniform gap to capacity, less than 1 dB at moderate block-length, over a wide range of rates.

A. *Edge growth*

Encoding and decoding of LDPC codes are typically developed with their Tanner graph representation. Edge growth algorithms, in particular Progressive Edge Growth (PEG) [7], are able to produce Tanner graphs with a good "girth," which relates to good minimum distance characteristics of the codes. Generally, such algorithms emphasize selection of graph connections that benefit the performance of message passing decoders. Finite graphs produced by edge growth according to asymptotically optimal degree distributions have exhibited a robust performance, especially for high-rate and short block-length codes, and are employed here for constructing the daughter code (the high-rate base code), as well as in motivating the extension technique described herein.

Specifically, the PEG algorithm sequentially and greedily assigns edges in the graph such that the resulting local girth (length of the shortest cycle involving a new edge) is maximized. Edges are assigned one-by-one in order of increasing variable-degree, and, if desired, according to a given check-degree distribution (otherwise, the check-degrees are concentrated around their mean-value as related to the variable-degree distribution and code-length). Other variations of edge growth algorithms emphasize cycle connectivity in choosing which edges to add. Cycles that are well-connected to the rest of the graph benefit from a better mix of uncorrelated information regarding their code-bits in message passing decoding.

The edge growth algorithm is readily modified to extend a base graph according to specific degree distributions. This is referred to as *constrained edge growth*, since the base graph places constraints on both check- and variable-degree distributions of subsequent extension graphs. In using edge growth for extension as such, edges are only added to variable-nodes that exhibit a degree increase, with some variable-nodes potentially receiving no new edges. Constrained edge growth is able to closely match optimal variable-degree distributions over a course of many rates. It is mainly due to finite block-length and check-degree constraints that a pure edge growth approach is insufficient for producing rate-compatible parity matrices of good performance. Thus, in the proposed design technique, both edge-growth, for its variable-degree flexibility, as well as parity splitting (or check splitting) [8], for exerting a level of control over the parity-degree distribution, are used to construct rate-compatible graphs.

B. *Parity splitting*

Check-irregular constructions (where both the check and variable node degrees are varied), though forming a larger class of irregular LDPC codes, tend to exhibit worse performance than check-regular constructions (in which all check nodes are of the same degree). Anecdotal evidence suggests that it is much easier to construct good check-regular graphs at finite block lengths since the burden of variable-irregularity (in terms of local girth) is evenly distributed amongst the parity nodes. Thus, as a guiding rule of thumb, check-degrees of the extension codes should be concentrated as close as possible around a certain desired average degree, namely $d_{opt}(r)$, which is monotone increasing in the rate and given by density evolution [2].

A parity check equation may be split into multiple parity equations by introducing new degree-two symbol nodes (see [8]). For example, suppose the set $A = \{x_0, \ldots, x_{d-1}\}$ represents code-bits involved in a degree-d parity constraint: $\sum_{x \in A} x = 0$. Then, letting x_d denote a new degree-two code symbol, the given parity equation is split into two: the first involving bits $A_1 \cup \{x_d\}$, and the second involving bits $A_2 \cup \{x_d\}$, where A_1 and A_2 are disjoint and $A = A_1 \cup A_2$. Thus, if the new parities have degrees d_1 and d_2, respectively, then $d_1 + d_2 = d + 2$ must hold. This operation increases the number of check constraints by one, creating the incremental redundancy bit, x_d, while preserving the base code structure (note that adding the new parity equations returns the original). A redundancy-bit produced by parity splitting may be computed with either of the resulting representations. Moreover, with the exception of a new degree-two code-bit, the variable-node degree distribution remains the same.

Parity splitting is a practical method for creating rate-compatible parity matrices, since large degree check nodes in the base graph are converted into multiple nodes of smaller degree in extending graphs. Further, parity splitting is essentially a rate-less technique (see [6]) since redundancy is produced at the bit-level. Yet, the technique offers no flexibility over the resulting variable-degree distribution, and is therefore incapable of producing rate-compatible codes with optimized degree distributions. Thus, a hybrid approach is proposed in this paper, in which edge-growth is utilized for creating good graphs with appropriate variable-node degree distributions, and parity splitting is utilized for concentrating the check-degrees as base codes are extended.

Example 1: A rate-1/2 parity-check matrix, compatible with a rate-4/5 and rate-2/3 code, is constructed with the preceding design approach. Figure 2 is a scatter-plot representation of the irregular LDPC Tanner graph obtained for an information block size of $k = 600$ bits. Columns in the Figure represent variable-nodes of the graph, and rows represent the check-nodes. Accordingly, dots indicate edges connecting code-bits to parity-constraints. The gray shaded region indicates that an edge has arisen when the incident parity-node is split, yielding the incident code-symbol. Any new code-symbol without an edge in the grey shaded region is given by edge growth, which is further constrained to be lower-triangular over the new-code symbols.

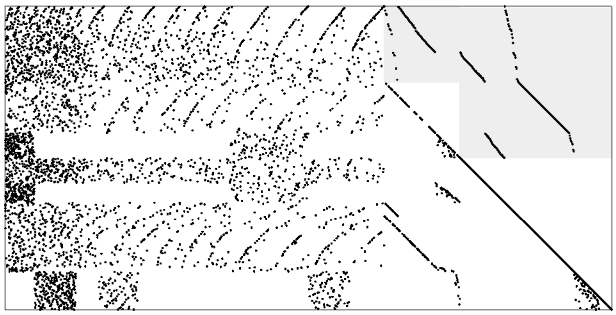

Fig. 2. Rate-1/2 parity-check matrix obtained by edge growth and parity splitting. Compatible with a rate-4/5 and rate-2/3 parity-check matrix, $k = 600$

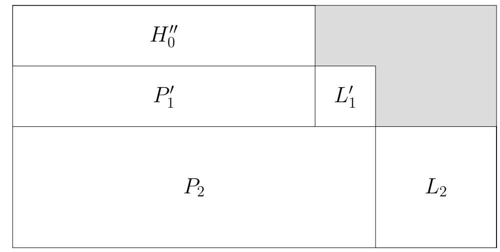

Fig. 3. Parity-check matrix decomposition for recursive encoding corresponding to example in Fig. 2

III. Encoding

The encoder is developed as a simple recursion, where extending code-words are are computed via matrix multiplication with base code-words. For this the following notation is used: The parity extending sub-matrix of the qth extension code is given by $[P_q\ L_q]$. In general, P_q is an $l_q \times n_{q-1}$ sparse matrix, where n_q denotes the length of the qth code and l_q denotes the number of new code-symbols, so that $l_q = n_q - n_{q-1}$. Similarly, L_q is an $l_q \times l_q$ sparse (and lower-triangular) matrix. Figure 3 illustrates the parity extending sub-matrix $[P_2\ L_2]$, corresponding to Example 1, where a rate-1/2 code extends a rate-2/3 base code.

Assuming L_q is invertible, and that rows of P_q are linearly independent, it is easy to show that $\mathbf{c}_q = \mathbf{c}_{q-1}[I_{n_{q-1}}\ (L_q^{-1}P_q)^{\mathrm{T}}]$ extends the base code-word $\mathbf{c}_{q-1}$ to code-word $\mathbf{c}_q$, where I_n denotes the $n \times n$ identity matrix. The extension algorithm developed constrains L_q to be lower-triangular and invertible (in fact L_q tends to be easily invertible, as observed for L_2 in Figure 2), and it is straight forward to solve $L_q^{-1}P_q$ by Gaussian elimination. Note that when the base graph is extended, any of its parity constraints are potentially split, thus the following nomenclature is adopted: Any sub-matrix X of the base graph becomes X' in the extending graph.

IV. Optimization framework

A framework for extending irregular LDPC codes to lower rates is described, and examples based on EXIT chart optimizations [12], [13], [14] are provided. Since EXIT charts rely on large code-word asymptotics, the optimization framework is essentially independent of information block-length, and thus one family of optimized degree distributions may be used to produce rate-compatible codes, for the same set of rates, for multiple information block-lengths.

Given a base graph of rate r_{n-1} and a target rate $r_n < r_{n-1}$, a fraction $\gamma_n = 1 - r_n/r_{n-1}$, relative to the extending code-word length, of new code symbols are introduced. The basic approach consists of two steps: (1) A certain fraction, denoted α_n, of the new code symbols are obtained by splitting check nodes of the base graph. This yields a fraction of $\alpha_n\gamma_n$ new degree-two variable nodes. (2) The remaining fraction, $1 - \alpha_n$, of new code symbols are developed by constrained edge growth. Thus, α_n and the extension graph check- and variable-degree distributions are the variables to be optimized. The optimization constraints are given by base graph check- and variable-degree distributions, and the extending rate, r_n.

All variable-degree distributions used in this paper are given by EXIT chart optimizations with a rate-compatible constraint. Then, since the base graph check-degree distribution is modified by splitting its check nodes, the focus is on choosing α_n and the degree-distribution of check-nodes produced by edge growth. Ideally, these parameters are optimized jointly, over the set of supported rates, in order to ensure a global performance characteristic. However, this would be computationally quite complex, and in the following we describe a simplified, sequential optimization procedure which yields an acceptable level of performance.

Simplified optimization: The rate-compatible codes are optimized sequentially, in order of decreasing rate. We assume that all parity nodes that are split are done so evenly, that they are split in order of largest degree, and that all new parity constraints developed by edge growth have the same, possibly fractional, degree, namely d_n. A fractional degree in this context is interpreted as an average degree arising from two consecutive integers. A heuristic which attempts to concentrate the check-degrees around their optimal mean-value, $d_{opt}(r_n)$, is used to choose α_n. Thus, with α_n and the base graph check-distribution specified, choosing $d_n \approx d_{opt}(r_n)$ suffices to describe the extending graph check-distribution, while adhering to the concentration heuristic. Finally, EXIT chart matching is employed to optimize the variable-degree distribution with afore mentioned constraints, including $\alpha_n\gamma_n$ new degree-two variable-nodes.

Figure 4 shows an example of EXIT charts of rate-compatible codes that results from the proposed optimization framework. The EXIT charts consist of variable- and check-node transfer functions that express an input-output mutual information relationship regarding the estimated code-symbols (see for example [13]). Starting from a daughter code and progressively optimizing the extending codes in order of decreasing rate places the most stringent constraints on lowest-rate code, and therefore a performance degradation is expected at low-rates. An optimization framework that employs reverse- as well as forward-compatibility constraints could be used to emphasize any member of the rate-compatible family.

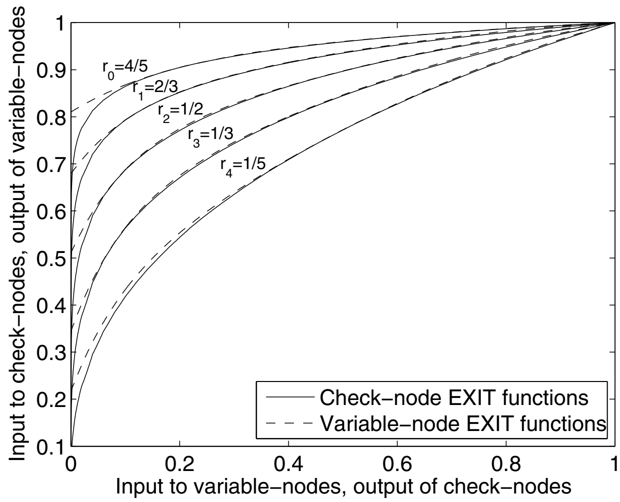

Fig. 4. Optimized EXIT functions with forward compatibility constraint

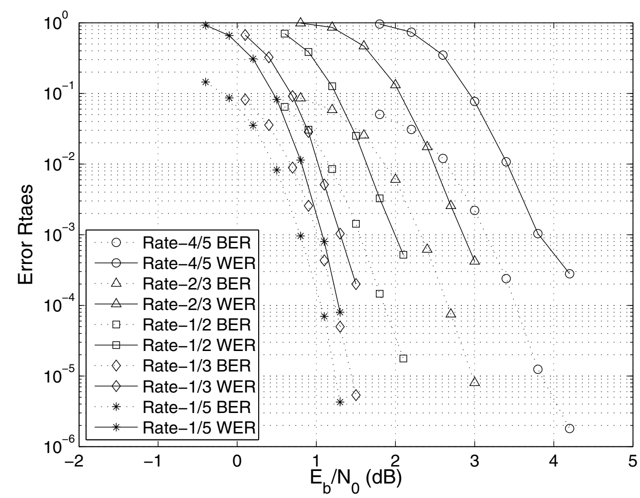

Fig. 5. Performance of rate-compatible irregular LDPC codes, $k = 600$

However, codes designed in this paper utilize the forward-compatible constrained optimization as described, which emphasizes the daughter code (i.e. the first transmission).

V. Results

Rate-compatible parity matrices are constructed according to the proposed design technique for the following set of rates: 4/5, 2/3, 1/2, 1/3, and 1/5. In this paper, the same set of optimized degree distributions is used to produce rate-compatible parity matrices for all information block-sizes. In constructing the codes, parity-nodes of largest degree are always split first, but otherwise in no particular order. The parity splitting technique benefits modestly by incorporating a cycle connectivity metric in choosing which nodes to split, especially for the high-rate codes. (Note that parity equations may be split in multiplicity which is useful if there is a significant step-size between the base- and extension-code rate.)

Figure 5 demonstrates the code-word error-rate (WER) and information-bit error-rate (BER) performance for an information block-size of $k = 600$ bits. The codes demonstrate a good performance for this block-size, as compared with turbo-code benchmarks, and exhibit no significant error floors up to WER of 1e-3.

Figure 6 shows the gap to capacity for information block-lengths of 600, 1500, and 6000, as measured at a BER of 1e-4. The results exhibit a roughly uniform gap to capacity, less than 1 dB at $k = 6000$, over a wide range of code rates. The gap to capacity begins to widen at low-rates, in the area of rate-1/5 for the design example provided. The widening gap at low code-rates stems from the simplified optimization technique, with forward compatibility constraint, as well as difficulties with conventional irregular LDPC designs at low code-rates.

VI. Conclusion

We demonstrate that extension based development of irregular LDPC Tanner graphs is able to produce rate-compatible codes of good performance. The technique is flexible in the range and granularity of the rates that it supports, and it inherently addresses the problem of puncturing which is present in mother code based designs. Moreover, since every subcode is viewed as puncture-less, this construction also shows a decoding complexity advantage.

As noted in [3], standard irregular LDPC code constructions are challenged at low code-rates. This issue has been addressed in the literature with the use of precoding techniques. Examples of pre-coded irregular codes on graphs include repeat accumulate (RA) style codes [4], [5], and Raptor codes [6]. Such architectures bear an increased similarity with turbo-codes, which perform well at low rates. With the additional constraint of forward-compatibility, it is conjectured that the application of precoding techniques could benefit the performance of rate-compatible LDPC codes built by extension. This is an item of future study. Work in this direction is reported in [15].

Codes designed in this paper differ significantly from the random-like constructions prescribed in [2]. Although asymptotic arguments are employed to optimize the degree-distributions, we further account for the specific matrix construction technique, which is chosen primarily to address finite block-length considerations. At the opposite end of the spectrum from large, random-looking graphs are proto-graph based constructions [16], which are derived from copies of a much smaller base graph, with highly structured inter-connections. Proto-graph based LDPC codes, in their simplicity (elegance), offer desirable implementation advantages. However, it is conjectured here that the reduced degrees of freedom of proto-graphs lead to an increased gap to capacity. This claim is supported by direct result comparisons with [17]. In short, the more “random-like” flexibility of extension-style constructions should benefit their performance, if at the cost increased complexity of description and implementation, but this is not yet quantified.

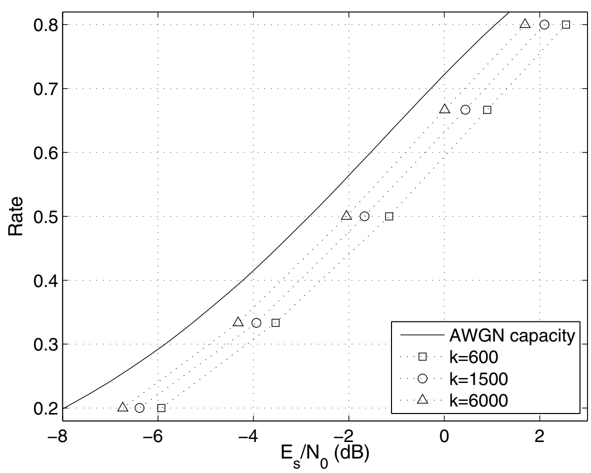

Fig. 6. Gap to capacity of the rate-compatible codes

References

[1] M. Luby, M. Mitzenmacher, A. Shokrollahi, and D. Spielman, "Improved low-density parity-check codes using irregular graphs," *IEEE Trans. Inform. Theory*, vol. 47, no. 2, pp. 585–598, Feb. 2001.

[2] 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.

[3] ——, "Multi-edge type LDPC codes," Apr. 2004, [Online]. Available: http://lthcwww.epfl.ch/papers/multiedge.ps.

[4] H. Jin, A. Khandekar, and R. McEliece, "Irregular repeat-accumulate codes," in *Proc. Intern. Symp. on Turbo Codes and Related Topics*, Sept. 2000.

[5] A. Abbasfar, D. Divsalar, and K. Yao, "Accumulate repeat accumulate codes," in *Proc. IEEE Intern. Symp. on Inform. Theory (ISIT)*, June 2004, p. 505.

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

[7] 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.

[8] M. Good and F. R. Kschischang, "Incremental redundancy via check splitting," in *Proc. 23rd Biennial Symp. on Commun.*, May 2006, pp. 55–58.

[9] H. Pishro-Nik and F. Fekri, "Results on punctured LDPC codes," in *IEEE Information Theory Workshop*, Oct. 2004, pp. 215–219.

[10] J. Li and K. R. Narayanan, "Rate-compatible low density parity check codes for capacity-approaching ARQ schemes in packet data communications," in *International Conference on Communications, Internet and Information Technology*, Nov. 2002, pp. 201–206.

[11] T. Tian and C. R. Jones, "Construction of rate-compatible LDPC codes utilizing information shortening and parity puncturing," *EURASIP Journal on Wireless Communications and Networking*, pp. 789–795, Oct. 2005.

[12] M. Tüchler and J. Hagenauer, "EXIT charts of irregular codes," in *Proc. Conf. on Inform. Sciences and Systems (CISS)*, Princeton, NJ, USA, Mar. 2002.

[13] S. ten Brink, G. Kramer, and A. Ashikhmin, "Design of low-density parity-check codes for modulation and detection," *IEEE Trans. Commun.*, vol. 52, no. 4, pp. 670–678, Apr. 2004.

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

[15] E. Soljanin, N. Varnica, and P. Whiting, "Incremental redundancy hybrid ARQ with LDPC and raptor codes," *IEEE Trans. Inform. Theory*, Sept. 2005, submitted for publication.

[16] S. Dolinar, "A rate-compatible family of protograph-based LDPC codes built by expurgation and lengthening," in *Proc. IEEE Intern. Symp. on Inform. Theory (ISIT)*, Sept. 2005, pp. 1627–1631.

[17] M. El-Khamy, J. Hou, and N. Bhushan, "H-ARQ rate-compatible structured LDPC codes," *Proc. IEEE Intern. Symp. on Inform. Theory (ISIT)*, pp. 1134–1138, July 2006.