Design Methods for Irregular Repeat Accumulate Codes

Aline Roumy[*], Souad Guemghar [†], Giuseppe Caire [‡], Sergio Verdú [§]

October 22, 2002

Abstract

We optimize the random-like ensemble of Irregular Repeat Accumulate (IRA) codes for binary-input symmetric channels in the large blocklength limit. Our optimization technique is based on approximating the Evolution of the Densities (DE) of the messages exchanged by the Belief-Propagation (BP) message-passing decoder by a one-dimensional dynamical system. In this way, the code ensemble optimization can be solved by linear programming. We propose four such DE approximation methods, and compare the performance of the obtained code ensembles over the binary symmetric channel (BSC) and the binary-antipodal input additive white Gaussian channel (BIAWGNC). Our results clearly identify the best among the proposed methods and show that the codes obtained by these methods are competitive with respect to the best-known irregular Low-Density Parity-Check codes (LDPC) codes, although both their design and their encoding/decoding are simpler.

1 Introduction

Since the discovery of Turbo codes [1], there have been several notable inventions in the field of random-like codes. In particular, the re-discovery of the LDPC codes, originally proposed in [2], the introduction of irregular LDPCs [3] and the introduction of the Repeat-Accumulate (RA) codes [4].

[*]INRIA, Rennes, France; aline.roumy@irisa.fr

[†]Insitut Eurécom, Sophia-Antipolis, France; souad.guemghar@eurecom.fr. The work of Souad Guemghar was supported in part by a PACA research grant.

[‡]Insitut Eurécom, Sophia-Antipolis, France; giuseppe.caire@eurecom.fr

[§]Dept. Electrical Engineering, Princeton University, USA; verdu@princeton.edu

In [3] irregular LDPCs were shown to asymptotically achieve the capacity of the binary erasure channel (BEC) under iterative message-passing decoding. Although the BEC is the only channel for which such a result currently exists, irregular LDPC codes have been designed for other binary-input channels (e.g., the BSC, the BIAWGNC [5], and the binary-input ISI channel [6, 7, 8]) and have shown to achieve very good performance.

First attempts to optimize irregular LDPC codes ([3] for the BEC and other channels [9]) used the DE technique that computes the expected performance for a random-like code ensemble in the limit of infinite code blocklength. In order to reduce the computational burden of ensemble optimization based on the DE, faster techniques have been proposed, based on the approximation of the DE by a one-dimensional dynamical system (recursion). These techniques are exact only for the BEC (for which DE is one-dimensional). The most popular techniques proposed so far are based on the Gaussian approximation (GA) of messages exchanged in the message passing decoder. GA in addition to the symmetry condition of message densities implies that the Gaussian density of messages is expressed by a single parameter. Techniques differ in the parameter to be tracked and in the mapping functions defining the dynamic system [10, 11, 12, 13, 14, 15, 16].

The introduction of irregular LDPCs motivated other schemes such as Irregular RA (IRA) [17], for which similar results exist (achievability of the BEC capacity) and Irregular Turbo codes [18]. IRA codes are in fact special subclasses of both irregular LDPCs and irregular Turbo codes. In IRA codes, a fraction f_i of information bits is repeated i times, for $i = 2, 3, ...$ The distribution $\{f_i \geq 0, i = 2, 3, ... \ : \ \sum_{i=2}^{\infty} f_i = 1\}$ is referred to as the *repetition profile*, and it is kept as a degree of freedom in the optimization of the IRA ensemble. After the repetition stage, the resulting sequence is interleaved and input to a recursive finite-state machine which outputs one bit for every a input symbols, where a is referred to as *grouping factor* and is also a design parameter.

IRA codes are an appealing choice because the encoder is extremely simple, their performance is quite competitive with that of Turbo codes and LDPCs, and they can be decoded with the lowest complexity of any iterative decoding scheme for random-like codes.

The only other work that has proposed a method to design IRA codes is [17, 19] where the design focuses on the choice of the grouping factor and the repetition profile. The recursive finite-state machine is the simplest one which gives full freedom to choose any rational number between 0 and 1 as the coding rate. We will also restrict our study to IRAs that use the same simple recursion of [17], although it might be expected that better codes can be obtained by including the finite-state machine as a degree of freedom in the overall ensemble optimization. The method used in [17] to choose the repetition

profile was based on the infinite-blocklength GA of message passing decoding proposed in [12]. In this work, we propose and compare four low-complexity ensemble optimization methods. Our approach to design IRAs is based on several tools that have been noticed recently: the EXtrinsic mutual Information Transfer (EXIT) function and its analytical properties [10, 20, 21], reciprocal channel (duality) approximation [22, 20], and the non-strict convexity of mutual information.

The rest of the paper is organized as follows. Section 2 presents the systematic IRA encoder and its related decoder: the BP message-passing algorithm. Existing results on the analysis of the decoder (i.e. DE technique) are summarized and applied to the IRA code-ensemble. This leads to a two-dimensional dynamical system whose state is defined on the space of symmetric distributions, for which we derive a local stability condition. In Section 3 we propose a general framework in order to approximate the DE (defined on the space of distributions) by a standard dynamical system defined on the reals. We propose four low-complexity ensemble optimization methods as special cases of our general framework. These methods differ by the way the message densities and the BP transformations are approximated:

1. GA, with reciprocal channel (duality) approximation;
2. BEC approximation, with reciprocal channel approximation;
3. GA, with EXIT function of the inner decoder;
4. BEC approximation, with EXIT function of the inner decoder.

All four methods lead to optimization problems solvable by linear programming. In Section 4 we show that the first proposed method yields a one-dimensional DE approximation with the same stability condition as the exact DE, whereas the exact stability condition must be added to the ensemble optimization as an explicit additional constraint for the second method. Then, we show that, in general, the GA methods are optimistic, in the sense that there is no guarantee that the optimized rate is below capacity. On the contrary, we show that for the BEC approximation methods rates below capacity are guaranteed. In Section 5 we compare our code optimization methods by evaluating their iterative decoding threshold (evaluated by the exact DE) over the BIAWGNC and the BSC.

2 Encoding, decoding and density evolution

Fig. 1 shows the block-diagram of a systematic IRA encoder. A block of information bits $\mathbf{b} = (b_1, \ldots, b_k) \in \mathbb{F}_2^k$ is encoded by an (irregular) repetition code of rate k/n. Each bit b_j is repeated r_j times, where $(r_1, \ldots, r_k)$ is a sequence of integers such that $2 \leq r_j \leq d$ and $\sum_{j=1}^{k} r_j = n$ (d is the maximum repetition factor). The block of repeated symbols is interleaved, and the resulting block $\mathbf{x}_1 = (x_{1,1}, \ldots, x_{1,n}) \in \mathbb{F}_2^n$ is encoded by an *accumulator*, defined by the recursion

$$x_{2,j+1} = x_{2,j} + \sum_{i=0}^{a-1} x_{1,aj+i}, \quad j = 0, \ldots, m-1 \tag{1}$$

with initial condition $x_{2,0} = 0$, where $\mathbf{x}_2 = (x_{2,1}, \ldots, x_{2,m}) \in \mathbb{F}_2^m$ is the accumulator output block corresponding to the input $\mathbf{x}_1$, $a \geq 1$ is a given integer (referred to as *grouping factor*), and we assume that $m = n/a$ is an integer. Finally, the codeword corresponding to the information block $\mathbf{b}$ is given by $\mathbf{x} = (\mathbf{b}, \mathbf{x}_2)$.

The transmission channel is memoryless, binary-input and symmetric-output, i.e., its transition probability $p_{Y|X}(y|x)$ satisfies

$$p_{Y|X}(y|0) = p_{Y|X}(-y|1) \tag{2}$$

where $y \mapsto -y$ indicates a *reflection* of the output alphabet. [1]

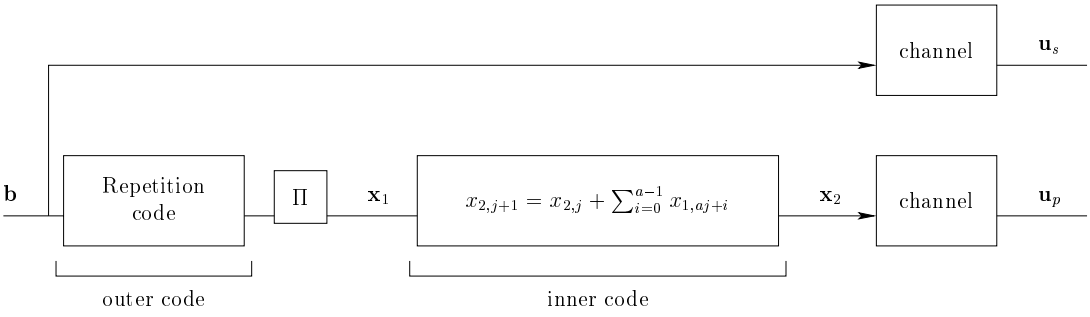

Figure 1: IRA encoder.

IRA codes are best represented by their Tanner graph [23] (see Fig. 2). In general, the Tanner graph of a linear code is a bipartite graph whose node set is partitioned into two subsets: the *bitnodes*, corresponding to the coded symbols, and the *checknodes*, corresponding to the parity-check equations that codewords must satisfy. The graph has

[1] If the output alphabet is the real line, then $-y$ coincides with ordinary reflection with respect to the origin. Generalizations to other alphabets are immediate.

an edge between bitnode α and checknode β if the symbol corresponding to α participates in the parity-check equation corresponding to β.

Since the IRA encoder is systematic (see Fig. 1), it is useful to further classify the bitnodes into two subclasses: the information bitnodes, corresponding to information bits, and the parity bitnodes, corresponding to the symbols output by the accumulator. Those information bits that are repeated i times are represented by bitnodes with degree i, as they participate in i parity-check equations. Each checknode is connected to a information bit nodes and to two parity bitnodes and represents one of the equations (for a particular j) (1). The connections between checknodes and information bitnodes are determined by the interleaver and are highly randomized. On the contrary, the connections between checknodes and parity bitnodes are arranged in a regular zig-zag pattern since, according to (1), every pair of consecutive parity bits are involved in one parity-check equation.

A random IRA code ensemble with parameters $(\{\lambda_i\}, a)$ and (information) blocklength k is formed by all graphs of the form of Fig. 2 with k information bitnodes, grouping factor a and $\lambda_i n$ edges connected to information bitnodes of degree i, for $i = 2, \ldots, d$. The sequence of non-negative coefficients $\{\lambda_i\}$ such that $\sum_{i=2}^{d} \lambda_i = 1$ is referred to as the *degree distribution* of the ensemble. The probability distribution over the code ensemble is induced by the uniform probability over all interleavers (permutations) of n elements.

The information bitnodes average degree is given by $\bar{d} \stackrel{\Delta}{=} 1/(\sum_{i=2}^{d} \lambda_i/i)$. The number of edges connecting information bitnodes to checknodes is $n = k/(\sum_{i=2}^{d} \lambda_i/i)$. The number of parity bitnodes is $m = k/(a \sum_{i=2}^{d} \lambda_i/i)$. Finally, the code rate is given by

$$R = \frac{k}{k+m} = \frac{a \sum_{i=2}^{d} \lambda_i/i}{1 + a \sum_{i=2}^{d} \lambda_i/i} \tag{3}$$

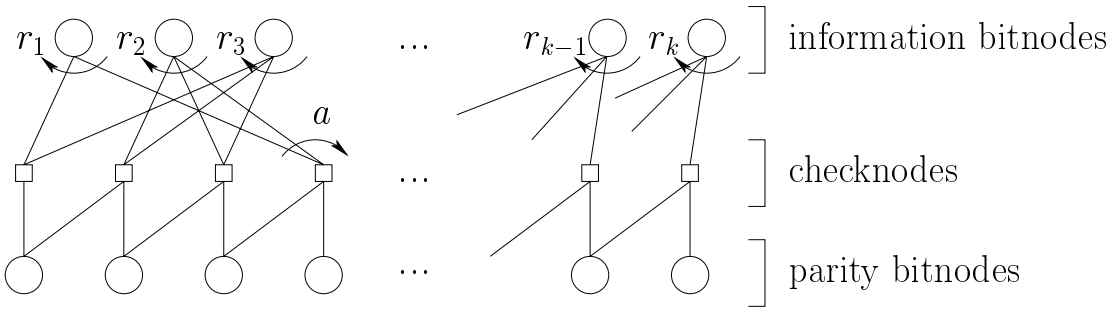

Figure 2: Tanner graph of an IRA code.

2.1 Belief propagation decoding of IRA codes

In this work we consider BP message-passing decoding [24, 25, 26]. In message-passing decoding algorithms, the graph nodes receive messages from their neighbors, compute new messages and forward them to their neighbors. The algorithm is defined by the code Tanner graph, by the set on which messages take on values, by the node computation rules and by the node activation scheduling.

In BP-decoding messages take on values in the extended real line $\mathbb{R} \cup \{-\infty, \infty\}$. The BP decoder is initialized by setting all messages output by the checknodes equal to zero. Each bitnode α is associated with the *channel observation* message (log-likelihood ratio)

$$u_\alpha = \log \frac{p_{Y|X}(y_\alpha | x_\alpha = 0)}{p_{Y|X}(y_\alpha | x_\alpha = 1)} \tag{4}$$

where y_α is the channel output corresponding to the transmission of the code symbol x_α.

The BP node computation rules are given as follows. For a given node we identify an adjacent edge as *outgoing* and all other adjacent edges as *incoming*. Consider a bitnode α of degree i and let $m_1, \ldots, m_{i-1}$ denote the messages received from the $i-1$ incoming edges and u_α the associated channel observation message. The message $m_{o,\alpha}$ passed along the outgoing edge is given by

$$m_{o,\alpha} = m_1 + \cdots + m_{i-1} + u_\alpha \tag{5}$$

Consider a checknode β of degree i and let $m_1, \ldots, m_{i-1}$ denote the messages received from the $i-1$ incoming edges. The message $m_{o,\beta}$ passed along the outgoing edge is given by

$$m_{o,\beta} = \gamma^{-1} \left(\gamma(m_1) + \cdots + \gamma(m_{i-1}) \right), \tag{6}$$

where the mapping $\gamma : \mathbb{R} \to \mathbb{F}_2 \times \mathbb{R}_+$ is defined by [9]

$$\gamma(z) = \left(\text{sign}(z), -\log \tanh \frac{|z|}{2} \right) \tag{7}$$

and where the sign function is defined as [9]

$$\text{sign}(z) = \begin{cases} 0 & \text{if } z > 0 \\ 0 & \text{with prob. } 1/2 \text{ if } z = 0 \\ 1 & \text{with prob. } 1/2 \text{ if } z = 0 \\ 1 & \text{if } z < 0 \end{cases}$$

Since the code Tanner graph has cycles, different schedulings yield in general non-equivalent BP algorithms. In this work we shall consider the following "classical" schedulings:

- LDPC-like scheduling [17]. In this case, all bitnodes and all checknodes are activated alternately and in parallel. Every time a node is activated, it sends outgoing messages to all its neighbors. A decoding iteration (or "round" [29]) consists of the activation of all bitnodes and all checknodes.

- Turbo-like scheduling. Following [27], a good decoding scheduling consists of isolating large trellis-like subgraphs (or, more generally, normal realizations in Forney's terminology) and applying locally the forward-backward BCJR algorithm [28] (that implements efficiently the BP algorithm on normal cycle-free graphs), as done for Turbo codes [1]. A decoding iteration consists of activating all the information bitnodes in parallel (according to (5)) and of running the BCJR algorithm over the entire accumulator trellis. In particular, the checknodes do not send messages to the information bitnodes until the BCJR iteration is completed.

Notice that for both of the above schedulings one decoder iteration corresponds to the activation of all information bitnodes in the graph exactly once.

2.2 Density evolution and stability

The BER performance of BP decoding averaged over the IRA code ensemble can be analyzed, for any finite number ℓ of iterations and in the limit of $k \to \infty$, by the DE technique [9].

For a given bitnode α and iteration ℓ, the message sent over an outgoing edge (say edge e) is a random variable that depends on the transmitted codeword, the channel noise and the interleaver (uniformly distributed over the set of permutations of n elements). The DE method finds the distribution of this random variable averaged over the channel noise and the interleaver, assuming that the blocklength goes to infinity. Under such assumption, the probability that an oriented neighborhood of depth 2ℓ of the edge e contains cycles vanishes. Therefore, DE can be computed under the cycle-free condition, implying that the input messages at any node in the BP algorithm are statistically independent. For binary-input symmetric-output channels, the average message distributions do not depend on the transmitted codeword [29], so the transmission of the all-zero codeword can be assumed.

The usefulness of the DE method stems from the *Concentration Theorem* [29, 3] which guarantees that, with high probability, the BER after ℓ iterations of the BP decoder applied to a randomly selected code in the ensemble and to a randomly generated channel noise sequence is close to the BER computed by DE with high probability, for sufficiently large blocklength.

Next, we formulate the DE for IRA codes and we study the stability condition of the fixed-point corresponding to zero BER. As in [9, section III-B], we introduce the space of *distributions* whose elements are non-negative non-decreasing right-continuous functions with range in $[0,1]$ and domain the extended real line.

It can be shown that, for a binary-input symmetric-output channel, the distributions of messages at any iteration of the DE satisfy the symmetry condition

$$\int h(x)dF(x) = \int e^{-x}h(-x)dF(x) \tag{8}$$

for any function h for which the integral exists. If F has density f, (8) is equivalent to

$$f(x) = e^x f(-x) \tag{9}$$

With some abuse of terminology, distributions satisfying (8) are said to be *symmetric*. The space of symmetric distributions will be denoted by $\mathcal{F}_{\rm sym}$.

The BER operator $\mathrm{Pe} : \mathcal{F}_{\rm sym} \to [0, 1/2]$ is defined by

$$\mathrm{Pe}(F) = \frac{1}{2}(F^-(0) + F(0))$$

where $F^-(z)$ is the left-continuous version of $F(z)$. We introduce the "delta at zero" distribution, denoted by Δ_0, for which $\mathrm{Pe}(\Delta_0) = 1/2$, and the "delta at infinity" distribution, denoted by Δ_∞, for which $\mathrm{Pe}(\Delta_\infty) = 0$.

The symmetry property (8) implies that a sequence of symmetric distributions $\{F^{(\ell)}\}_{\ell=0}^{\infty}$ converges to Δ_∞ if and only if $\lim_{\ell\to\infty} \mathrm{Pe}(F^{(\ell)}) = 0$, where convergence of distributions is in the sense given in [9, Sect. III-F].

The DE for IRA code ensembles is given by the following proposition whose derivation is omitted as it is completely analogous to the derivation of DE in [9] for irregular LDPC codes.

Proposition 1. Let P_ℓ [resp., $\widetilde{P}_\ell$] denote the average distribution of messages passed from an information bitnode [resp., parity bitnode] to a checknode, at iteration ℓ. Let Q_ℓ [resp., $\widetilde{Q}_\ell$] denote the average distribution of messages passed from a checknode to an information bitnode [resp., parity bitnode], at iteration ℓ.

Under the cycle-free condition, $P_\ell, \widetilde{P}_\ell, Q_\ell, \widetilde{Q}_\ell$ satisfy the following recursion:

$$\begin{aligned}
P_\ell &= F_u \otimes \lambda(Q_\ell) && (10)\\
\widetilde{P}_\ell &= F_u \otimes \widetilde{Q}_\ell && (11)\\
Q_\ell &= \Gamma^{-1}\left(\Gamma(\widetilde{P}_{\ell-1})^{\otimes 2} \otimes \Gamma(P_{\ell-1})^{\otimes(a-1)}\right) && (12)\\
\widetilde{Q}_\ell &= \Gamma^{-1}\left(\Gamma(\widetilde{P}_{\ell-1}) \otimes \Gamma(P_{\ell-1})^{\otimes a}\right) && (13)
\end{aligned}$$

for $\ell = 1, 2, \ldots$, with initial condition $P_0 = \widetilde{P}_0 = \Delta_0$, where F_u denotes the distribution of the channel observation messages (4), $^{\otimes}$ denotes convolution of distributions, defined by

$$(F \otimes G)(z) = \int F(z-t)dG(t) \tag{14}$$

$^{\otimes m}$ denotes m-fold convolution, $\lambda(F) \stackrel{\Delta}{=} \sum_{i=2}^{d} \lambda_i F^{\otimes(i-1)}$, $\Gamma(F_x)$ is the distribution of $y = \gamma(x)$ (defined on $\mathbb{F}_2 \times \mathbb{R}_2$), when $x \sim F_x$, and Γ^{-1} denotes the inverse mapping of Γ, i.e., $\Gamma^{-1}(G_y)$ is the distribution of $x = \gamma^{-1}(y)$ when $y \sim G_y$. □

The DE recursion (10 – 13) is a two-dimensional non-linear dynamical system with state-space $\mathcal{F}^2_{\text{sym}}$ (i.e., the state trajectories of (10 – 13) are sequences of pairs of symmetric distributions $(P_\ell, \widetilde{P}_\ell)$). For this system, the BER at iteration ℓ is given by $\text{Pe}(P_\ell)$.

It is easy to see that $(\Delta_\infty, \Delta_\infty)$ is a fixed-point of (10 – 13). The local stability of this fixed-point is given by the following result:

Theorem 1. The fixed-point $(\Delta_\infty, \Delta_\infty)$ for the DE is locally stable if and only if

$$\lambda_2 < \frac{e^r(e^r - 1)}{a + 1 + e^r(a-1)} \tag{15}$$

where $r = -\log(\int e^{-z/2} dF_u(z))$.

Proof. See Appendix A.1. □

Here necessity and sufficiency are used in the sense of [9]. By following steps analogous to [9], it can be shown that if (15) holds, then there exists $\xi > 0$ such that if for some $\ell \in \mathbb{N}$, $\text{Pe}(RP_\ell(P_0, \widetilde{P}_0) + (1-R)\widetilde{P}_\ell(P_0, \widetilde{P}_0)) < \xi$ then $\text{Pe}(RP_\ell + (1-R)\widetilde{P}_\ell)$ converges to zero as ℓ tends to infinity. On the contrary, if λ_2 is strictly larger than the RHS of (15), then there exists $\xi > 0$ such that for all $\ell \in \mathbb{N}$ $\text{Pe}(RP_\ell(P_0, \widetilde{P}_0) + (1-R)\widetilde{P}_\ell(P_0, \widetilde{P}_0)) > \xi$.

3 IRA ensemble optimization

In this section we tackle the problem of optimizing the IRA code ensemble parameters for a broad class of binary-input symmetric-output channels.

A property of DE given in Proposition 1 is that $\text{Pe}(P_\ell)$ for $\ell = 1, 2, \ldots$ is a non-increasing non-negative sequence. Hence, the limit $\lim_{\ell\to\infty} \text{Pe}(P_\ell)$ exists. Consider a family of channels $\mathcal{C}(\nu) = \{p^\nu_{Y|X} : \nu \in \mathbb{R}_+\}$, where the channel parameter ν is, for example, an indicator of the noise level in the channel. Following [29], we say that $\mathcal{C}(\nu)$ is monotone with respect to the IRA code ensemble $(\{\lambda_i\}, a)$ under BP-decoding if, for

any finite ℓ, $\nu \leq \nu' \Leftrightarrow \mathrm{Pe}(P_\ell) \leq \mathrm{Pe}(P'_\ell)$, where P_ℓ and P'_ℓ are the message distributions at iteration ℓ of DE applied to channels $p^{\nu}_{Y|X}$ and $p^{\nu'}_{Y|X}$, respectively.

Let $\mathrm{BER}(\nu) = \lim_{\ell\to\infty} \mathrm{Pe}(P_\ell)$, where $\{P_\ell\}$ is the trajectory of DE applied to the channel $p^{\nu}_{Y|X}$. The *threshold* $\nu^\star$ of the ensemble $(\{\lambda_i\}, a)$ over the monotone family $\mathcal{C}(\nu)$ is the worst-case channel parameter for which the limiting BER is zero, i.e.,

$$\nu^\star = \sup\{\nu \geq 0 \ : \ \mathrm{BER}(\nu) = 0\} \tag{16}$$

Thus, for every value of ν, the optimal IRA ensemble parameters a and $\{\lambda_i\}$ maximize R subject to vanishing $\mathrm{BER}(\nu) = 0$, i.e., are solution of the optimization problem

$$\begin{cases} \text{maximize} & a \sum_{i=2}^{d} \lambda_i / i \\ \text{subject to} & \sum_{i=2}^{d} \lambda_i = 1, \quad \lambda_i \geq 0 \ \forall \ i \\ \text{and to} & \mathrm{BER}(\nu) = 0 \end{cases} \tag{17}$$

the solution of which can be found by some numerical techniques, as in [9]. However, the constraint $\mathrm{BER}(\nu) = 0$ is given directly in terms of the fixed-point of the DE recursion, and makes optimization computationally very intensive.

A variety of methods have been developed in order to simplify the code ensemble optimization [17, 22, 12, 30]. They consist of replacing the DE with a dynamical system defined over the reals (rather than over the space of distributions), whose trajectories and fixed-points are related in some way to the trajectories and fixed-point of the DE. Essentially, all proposed approximated DE methods can be formalized as follows. Let $\Phi : \mathcal{F}_{\mathrm{sym}} \to \mathbb{R}$ and $\Psi : \mathbb{R} \to \mathcal{F}_{\mathrm{sym}}$ be mappings of the set of symmetric distributions to the real numbers and viceversa. Then, a dynamical system with state-space $\mathbb{R}^2$ can be derived from (10 – 13) as

$$\begin{aligned} x_\ell &= \Phi\left(F_u \otimes \lambda\left(\mathsf{Q}_\ell\right)\right) && (18)\\ \widetilde{x}_\ell &= \Phi\left(F_u \otimes \widetilde{\mathsf{Q}}_\ell\right) && (19)\\ \mathsf{Q}_\ell &= \Gamma^{-1}\left(\Gamma\left(\Psi(\widetilde{x}_{\ell-1})\right)^{\otimes 2} \otimes \Gamma\left(\Psi(x_{\ell-1})\right)^{\otimes(a-1)}\right) && (20)\\ \widetilde{\mathsf{Q}}_\ell &= \Gamma^{-1}\left(\Gamma\left(\Psi(\widetilde{x}_{\ell-1})\right) \otimes \Gamma\left(\Psi(x_{\ell-1})\right)^{\otimes a}\right) && (21) \end{aligned}$$

for $\ell = 1, 2, \ldots$, with initial condition $x_0 = \widetilde{x}_0 = \Phi(\Delta_0)$, and where $(x_\ell, \widetilde{x}_\ell)$ are the system state variables.

By eliminating the intermediate distributions Q_ℓ and $\widetilde{\mathsf{Q}}_\ell$, we can put (18 – 21) in the form

$$\begin{aligned} x_\ell &= \phi(x_{\ell-1}, \widetilde{x}_{\ell-1}) \\ \widetilde{x}_\ell &= \widetilde{\phi}(x_{\ell-1}, \widetilde{x}_{\ell-1}) \end{aligned} \tag{22}$$

For all DE approximations considered in this work, the mappings Φ and Ψ and the functions ϕ and $\widetilde{\phi}$ satisfy the following desirable properties:

1. $\Phi(\Delta_0) = 0$, $\Phi(\Delta_\infty) = 1$.
2. $\Psi(0) = \Delta_0$, $\Psi(1) = \Delta_\infty$.
3. ϕ and $\widetilde{\phi}$ are defined on $[0,1] \times [0,1]$ and have range in $[0,1]$.
4. $\phi(0,0) > 0$ and $\widetilde{\phi}(0,0) > 0$.
5. $\phi(1,1) = \widetilde{\phi}(1,1) = 1$, i.e., $(1,1)$ is a fixed-point of the recursion (22). Moreover, this fixed-point corresponds to the zero-BER fixed-point $(\Delta_\infty, \Delta_\infty)$ of the exact DE.
6. The function $\widetilde{\phi}(x, \widetilde{x})$ is monotonically increasing in $\widetilde{x}$ for all $x \in [0,1]$, $\widetilde{\phi}(x,0) \geq 0$ and $\widetilde{\phi}(x,1) \leq 1$. Therefore, the equation

$$\widetilde{x} = \widetilde{\phi}(x, \widetilde{x})$$

has a unique solution in $[0,1]$ for all $x \in [0,1]$. This solution will be denoted by $\widetilde{x}(x)$.

It follows that all fixed points of (22) must satisfy

$$x = \phi(x, \widetilde{x}(x)) \tag{23}$$

and that in order to avoid fixed-points other than $(1,1)$, (23) must not have solutions in the interval $[0,1)$, i.e., it must satisfy

$$x < \phi(x, \widetilde{x}(x)), \quad \forall \ x \in [0,1) \tag{24}$$

Notice that, in general, (24) is neither a necessary nor a sufficient condition for the uniqueness of the zero-BER fixed-point of the exact DE. However, if the quality of the DE approximation is good, this provides a heuristic for the code ensemble optimization.

By replacing the constraint $\mathrm{BER}(\nu) = 0$ by (24) in (17), we obtain the *approximated* IRA ensemble optimization method as

$$\begin{cases} \text{maximize} & a \sum_{i=2}^{d} \lambda_i / i \\ \text{subject to} & \sum_{i=2}^{d} \lambda_i = 1, \quad \lambda_i \geq 0 \ \forall \ i \\ \text{and to} & x < \phi(x, \widetilde{x}(x)), \quad \forall \ x \in [0,1) \end{cases} \tag{25}$$

Approximations of the DE recursion differ essentially in the choice of Φ and Ψ, and in the way the *intermediate* distributions Q_ℓ and $\widetilde{\mathsf{Q}}_\ell$ and the channel message distribution F_u are approximated. Next, we illustrate the approximation methods considered in this work.

3.1 EXIT functions

Several recent works show that DE can be accurately described in terms of the evolution of the mutual information between the variables associated with the bitnodes and their messages (see [10, 31, 11, 32, 21, 33, 16]).

The key idea in order to approximate DE by mutual information evolution is to describe each computation node in BP-decoding by a *mutual information transfer function*. For historical reasons, this function is usually referred to as the EXtrinsic mutual Information Transfer (EXIT) function.

EXIT functions are generally defined as follows. Consider the model of Fig. 3, where the box represents a generalized computation node of the BP algorithm (i.e., it might contain a subgraph formed by several nodes and edges, and might depend on some other random variables such as channel observations, not shown in Fig. 3). Let $m_1, \ldots, m_{i-1}$ denote the input messages, assumed i.i.d. $\sim F_{\rm in}$, and let $m_o \sim F_{\rm out}$ denote the output message. Let X_j denote the binary code symbol associated with message m_j, for $j = 1, \ldots, i-1$, and let X denote the binary code symbol associated with message m_o. Since $F_{\rm in}, F_{\rm out} \in \mathcal{F}_{\rm sym}$, we can think of m_j and m_o as the outputs of binary-input symmetric-output channels with inputs X_j and X and transition probabilities

$$P(m_j \leq z | X_j = 0) = F_{\rm in}(z) \quad (26)$$

$$P(m_o \leq z | X = 0) = F_{\rm out}(z), \quad (27)$$

respectively.

Channel (26) models the *a priori* information that the node receives about the symbols X_j's, and the channel (27) models the *extrinsic information* [1] that the node generates about the symbol X.

We define the binary-input symmetric-output capacity functional $\mathcal{I} : \mathcal{F}_{\rm sym} \to [0,1]$, such that

$$\mathcal{I}(F) = 1 - \int_{-\infty}^{\infty} \log_2 \left(1 + e^{-z}\right) dF(z) \quad (28)$$

Namely, $\mathcal{I}$ maps any symmetric distribution F into the capacity [2] of the binary-input symmetric-output channel with transition probability $p_{Y|X}(y|0) = F(y)$.

Then, we let

$$I_A = I(X_j; m_j) = \mathcal{I}(F_{\rm in})$$

$$I_E = I(X; m_o) = \mathcal{I}(F_{\rm out})$$

[2] Recall that the capacity of a binary-input symmetric-output memoryless channel is achieved by uniform i.i.d. inputs.

denote the capacities of the channels (26) and (27), respectively. The EXIT function of the node of Fig. 3 is the set of pairs (I_A, I_E), for all $I_A \in [0,1]$ and for some (arbitrary) choice of the input distribution F_{in} such that $\mathcal{I}(F_{\text{in}}) = I_A$. Notice that the EXIT function of a node is not uniquely defined, since it depends on the choice of F_{in}. In general, different choices yield different transfer functions.

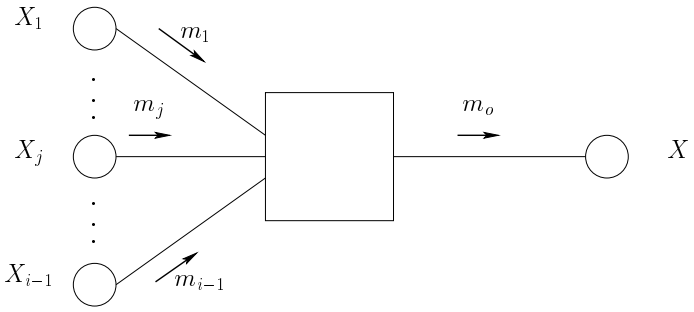

Figure 3: EXIT model.

The approximations of the DE considered in this work are based on EXIT functions, and track the evolution of the mutual information between the messages output by the bitnodes and the associated code symbols.

Remark: Two properties of binary-input symmetric-output channels. Before concluding this section, we take a brief detour in order to point out two properties of binary-input symmetric-output channels. Consider a binary-input symmetric-output channel with $p_{Y|X}(y|0) = G(y)$, where G is not necessarily symmetric (in the sense of (8)). Its capacity can be written as

$$C = 1 - \int_{-\infty}^{\infty} \log_2 \left(1 - \frac{dG(-z)}{dG(z)}\right) dG(z) \tag{29}$$

By concatenating the transformation $y \mapsto u = \log \frac{p_{Y|X}(y|0)}{p_{Y|X}(y|1)}$ to the channel output, we obtain a new binary-input symmetric-output channel with $p'_{U|X}(u|0) = F(u)$ such that $F \in \mathcal{F}_{\text{sym}}$. Moreover, since U is a sufficient statistics for Y, the original channel has the same capacity as the new channel, given by $C = \mathcal{I}(F)$. Therefore, by defining appropriately the channel output, the capacity of any binary-input symmetric-output channel can always be put in the form (28).

Another interesting property is the following:

Proposition 2. The mutual information functional is not strictly convex on the set of binary-input symmetric-output channels with transition probability $p_{Y|X}(y|0) \in \mathcal{F}_{\text{sym}}$.

Proof. See Appendix A.2. □

3.2 Method 1

The first approximation of the DE considered in this work assumes that the distributions at any iteration are Gaussian. A Gaussian distribution satisfies the symmetry condition (9) if and only if its variance is equal to twice the absolute value of its mean. We introduce the short-hand notation $\mathcal{N}_{\text{sym}}(\mu)$ to denote the symmetric Gaussian distribution (or density, depending on the context) with mean μ, i.e., $\mathcal{N}_{\text{sym}}(\mu) \triangleq \mathcal{N}(\mu, 2|\mu|)$.

For a distribution $F \in \mathcal{F}_{\text{sym}}$, we let the mapping Φ be equal to $\mathcal{I}$ defined in (28), and for all $x \in [0,1]$ we define the mapping

$$\Psi : x \mapsto \mathcal{N}_{\text{sym}}(J^{-1}(x)) \tag{30}$$

where

$$J(\mu) \triangleq \mathcal{I}(\mathcal{N}_{\text{sym}}(\mu)) = 1 - \frac{1}{\sqrt{\pi}} \int_{-\infty}^{+\infty} e^{-z^2} \log_2\left(1 + e^{-2\sqrt{\mu}z - \mu}\right) dz, \tag{31}$$

Namely, Ψ maps $x \in [0,1]$ into the symmetric Gaussian distribution $\mathcal{N}_{\text{sym}}(\mu)$ such that the BIAWGNC with transition probability $p_{Y|X}(y|0) = \mathcal{N}_{\text{sym}}(\mu)$ has capacity x.

The first key approximation in Method 1 is

$$\begin{aligned} \mathsf{Q}_\ell &\approx \mathcal{N}_{\text{sym}}(\mu_\ell) \\ \widetilde{\mathsf{Q}}_\ell &\approx \mathcal{N}_{\text{sym}}(\widetilde{\mu}_\ell) \end{aligned} \tag{32}$$

for some $\mu_\ell, \widetilde{\mu}_\ell \geq 0$.

In order to compute μ_ℓ and $\widetilde{\mu}_\ell$ we make use of the reciprocal channel approximation [22] also called *approximate* duality property of EXIT functions in [20]. This states that the EXIT function of a checknode is accurately approximated by the EXIT function of a bitnode with the same degree after the change of variables $I_A \mapsto 1 - I_A$ and $I_E \mapsto 1 - I_E$ (see Fig. 4). Using approximate duality, we replace the checknode by a bitnode and

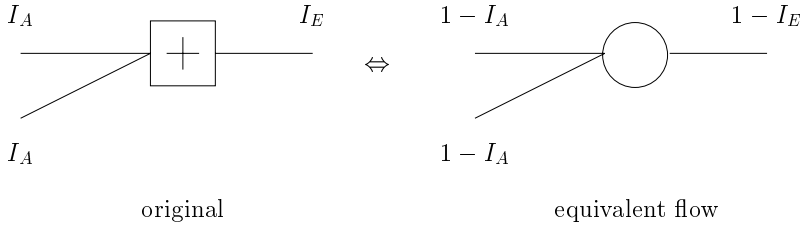

Figure 4: Reciprocal channel approximation.

change $(x_{\ell-1}, \widetilde{x}_{\ell-1})$ into $(1-x_{\ell-1}, 1-\widetilde{x}_{\ell-1})$. Since for a bitnode the output message is the sum of the input messages (see (5)), and since the input distributions $\Psi(1-x_{\ell-1})$ and $\Psi(1-\widetilde{x}_{\ell-1})$) are Gaussian, also the output distribution is Gaussian, with mean

$$(a-1)J^{-1}(1-x_{\ell-1}) + 2J^{-1}(1-\widetilde{x}_{\ell-1})$$

for messages sent to information bitnodes and

$$aJ^{-1}(1-x_{\ell-1}) + J^{-1}(1-\widetilde{x}_{\ell-1})$$

for messages sent to parity bitnodes. Finally, μ_ℓ and $\widetilde{\mu}_\ell$ are given by

$$\begin{aligned} \mu_\ell &= J^{-1}\left(1 - J\left((a-1)J^{-1}(1-x_{\ell-1}) + 2J^{-1}(1-\widetilde{x}_{\ell-1})\right)\right) \\ \widetilde{\mu}_\ell &= J^{-1}\left(1 - J\left(aJ^{-1}(1-x_{\ell-1}) + J^{-1}(1-\widetilde{x}_{\ell-1})\right)\right) \end{aligned} \tag{33}$$

The second key approximation in Method 1 is to replace F_u with a discrete (symmetric) distribution such that

$$F_u \approx \sum_{j=1}^{D} p_j \Delta_{v_j} \tag{34}$$

for some integer $D \geq 2$, $v_j \in \mathbb{R}$ and $p_j \in \mathbb{R}_+$ such that $\sum_{j=1}^{D} p_j = 1$.

With this assumption, from the definition (28) of the operator $\mathfrak{I}$ and since [9]: a) the convolution of symmetric distributions is symmetric, and b) the convex combination of symmetric distributions is symmetric, it is immediate to write (18) and (19) as

$$\begin{aligned} x_\ell &= 1 - \sum_{i=2}^{d}\sum_{j=1}^{D} \lambda_i p_j \frac{1}{\sqrt{\pi}} \int_{-\infty}^{\infty} e^{-z^2} \log_2\left(1 + e^{-2\sqrt{(i-1)\mu_\ell}z - (i-1)\mu_\ell - v_j}\right) dz \\ \widetilde{x}_\ell &= 1 - \sum_{j=1}^{D} p_j \frac{1}{\sqrt{\pi}} \int_{-\infty}^{\infty} e^{-z^2} \log_2\left(1 + e^{-2\sqrt{\widetilde{\mu}_\ell}z - \widetilde{\mu}_\ell - v_j}\right) dz \end{aligned} \tag{35}$$

The desired DE approximation in the form (22) is obtained (implicitly) by combining (33) and (35). Notice that (35) is linear in the repetition profile and the optimization problem (25) can be solved as linear programming.

Example 1: discrete-output channels. In general, when the channel output is discrete then the approximation (34) holds exactly. For example, for the BSC with transition probability p we have

$$F_u = p\Delta_{-\log\frac{1-p}{p}} + (1-p)\Delta_{\log\frac{1-p}{p}}$$

$\Diamond$

Example 2: The BIAWGNC defined by $y = (-1)^x + z$, where $z \sim \mathcal{N}(0, \sigma^2)$, is a channel such that

$$F_u = \mathcal{N}_{\text{sym}}(2/\sigma^2) \tag{36}$$

In this case, since convolving symmetric Gaussian distributions yields a symmetric Gaussian distribution whose mean is the sum of the means, the discretization approximation (34) is not necessary and we have

$$\begin{aligned} F_u \otimes \lambda(\mathsf{Q}_\ell) &= \sum_{i=2}^{d} \lambda_i \mathcal{N}_{\text{sym}}(2/\sigma^2 + (i-1)\mu_\ell) \\ F_u \otimes \widetilde{\mathsf{Q}}_\ell &= \mathcal{N}_{\text{sym}}(2/\sigma^2 + \widetilde{\mu}_\ell) \end{aligned} \tag{37}$$

By applying the operator $\mathfrak{I}$ and using (31) we obtain the DE approximation for the BIAWGNC as

$$\begin{aligned} x_\ell &= \sum_{i=2}^{d} \lambda_i J\left(\frac{2}{\sigma^2} + (i-1)J^{-1}\left(1 - J\left((a-1)J^{-1}(1-x_{\ell-1}) + 2J^{-1}(1-\widetilde{x}_{\ell-1})\right)\right)\right) \\ \widetilde{x}_\ell &= J\left(\frac{2}{\sigma^2} + J^{-1}\left(1 - J\left(aJ^{-1}(1-x_{\ell-1}) + J^{-1}(1-\widetilde{x}_{\ell-1})\right)\right)\right) \end{aligned} \tag{38}$$

$\Diamond$

3.3 Method 2

The second approximation of the DE considered in this work assumes that the distributions of messages at any iteration consist of two mass points, one at zero and the other at $+\infty$. For such distributions, we introduce the short-hand notation $\mathcal{E}_{\text{sym}}(\epsilon) \triangleq \epsilon\Delta_0 + (1-\epsilon)\Delta_\infty$.

We let the mapping Φ be equal to $\mathfrak{I}$ defined in (28) and the mapping Ψ be

$$\Psi : x \mapsto \mathcal{E}_{\text{sym}}(1-x) \tag{39}$$

for all $x \in [0, 1]$.

With these mappings, (20 – 21) can be put in the form

$$\begin{aligned} \mathsf{Q}_\ell &= \mathcal{E}_{\text{sym}}(1 - x_{\ell-1}^{a-1}\widetilde{x}_{\ell-1}^2) \\ \widetilde{\mathsf{Q}}_\ell &= \mathcal{E}_{\text{sym}}(1 - x_{\ell-1}^{a}\widetilde{x}_{\ell-1}) \end{aligned} \tag{40}$$

where we used the fact that, as it can be easily seen from the definitions of Γ and Γ^{-1} in (46 – 48),

$$\Gamma^{-1}\left(\Gamma(\mathcal{E}_{\text{sym}}(\epsilon_1)) \otimes \Gamma(\mathcal{E}_{\text{sym}}(\epsilon_2))\right) = \mathcal{E}_{\text{sym}}(1 - (1-\epsilon_1)(1-\epsilon_2))$$

Notice that, while in Method 1 we *assumed* Q_ℓ and $\widetilde{\mathsf{Q}}_\ell$ to be symmetric Gaussian (see (32)), here (40) holds exactly.

As a consequence of these mappings, the communication channel of the parity bits, with distribution F_u, is replaced by a BEC with erasure probability $\epsilon = 1 - \mathcal{I}(F_u)$.

Furthermore, for any $F \in \mathcal{F}_{\text{sym}}$ we have

$$\mathcal{I}(F \otimes \mathcal{E}_{\text{sym}}(\epsilon)) = 1 - (1 - \mathcal{I}(F))\epsilon$$

¿From this result, it is immediate to obtain the approximated DE recursion as

$$\begin{aligned} x_\ell &= 1 - (1 - \mathcal{I}(F_u)) \textstyle\sum_{i=2}^{d} \lambda_i \left(1 - x_{\ell-1}^{a-1} \widetilde{x}_{\ell-1}^2\right)^{i-1} \\ \widetilde{x}_\ell &= 1 - (1 - \mathcal{I}(F_u)) \left(1 - x_{\ell-1}^{a} \widetilde{x}_{\ell-1}\right) \end{aligned} \tag{41}$$

Notice that (41) is the standard (exact) DE for the IRA ensemble $(\{\lambda_i\}, a)$ over a BEC (see [17]) with the same capacity of the actual binary-input symmetric-output channel, given by $\mathcal{I}(F_u)$. We point out here that this method, consisting of replacing the actual channel with a BEC with equal capacity and optimizing the code ensemble for the BEC, was proposed in [22] for the optimization of LDPC ensembles. Interestingly, this method follows as a special case of our general approach for DE approximation, for a particular choice of the mappings Φ and Ψ.

In this case, the fixed-point equation corresponding to (23) is obtained in closed form as

$$x = 1 - (1 - \mathcal{I}(F_u)) \sum_{i=2}^{d} \lambda_i \left(1 - \frac{x^{a-1} \mathcal{I}(F_u)^2}{(1 - (1 - \mathcal{I}(F_u))x^a)^2}\right)^{i-1} \tag{42}$$

(for the details, see [17]).

3.4 Methods 3 and 4

Methods 1 and 2 yield (almost) closed-form DE approximations at the price of some approximations of the message distributions and, above all, of the checknodes output distributions Q_ℓ and $\widetilde{\mathsf{Q}}_\ell$.

In much of the current literature on randomlike code ensemble optimization, the EXIT function of a decoding block is obtained by Monte Carlo simulation, by generating i.i.d. input messages, estimating the distribution of the output messages and computing a one-dimensional quantity [10, 11, 12, 13, 14, 15, 16]. Following this approach, we shall consider the IRA decoder with turbo-like scheduling (see Fig. 5) and obtain the EXIT functions of the inner and outer decoders.

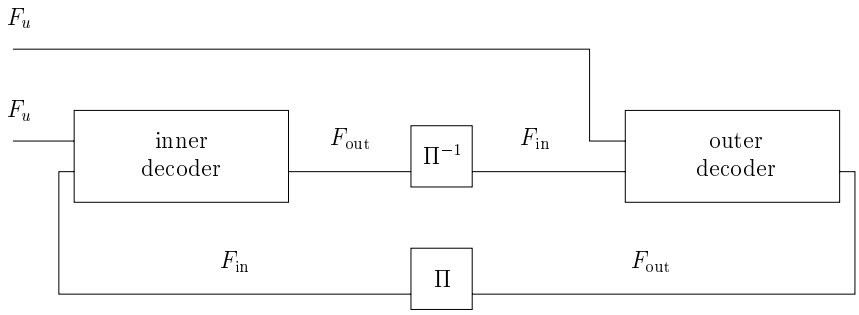

Figure 5: Turbo-like IRA decoder.

The inner (accumulator) and outer (repetition) decoders are characterized by an EXIT function as defined in Section 3.1, for some guess of the (symmetric) distribution F_{in}. In general, the EXIT function of the decoders can be obtained as follows:

1. Let the channel observation messages be i.i.d., $\sim F_u$.
2. Assume the decoder input messages are i.i.d., $\sim F_{\text{in}}$.
3. Obtain either in closed form or by Monte Carlo simulation the corresponding marginal distribution F_{out} of the decoder output messages.
4. Let $I_A = \mathcal{I}(F_{\text{in}})$, $I_E = \mathcal{I}(F_{\text{out}})$ be a point on the EXIT function curve.

Our Methods 3 and 4 consist of applying the above approach under the assumptions $F_{\text{in}} = \mathcal{N}_{\text{sym}}(J^{-1}(I_A))$ and $F_{\text{in}} = \mathcal{E}_{\text{sym}}(1 - I_A)$, respectively.

Let the resulting EXIT functions of the inner and outer decoders be denoted by $I_E = g(I_A)$ and by $I_E = h(I_A)$, respectively, and let x denote the mutual information between the messages at the output of the outer decoder (repetition code) and the corresponding symbols (information bitnodes).

The resulting approximated DE is given by

$$x_\ell = h(g(x_{\ell-1})) \tag{43}$$

The corresponding fixed-point equation is given by $x = h(g(x))$, and the condition for the uniqueness of the fixed-point at $x = 1$, corresponding to (24), is $x < h(g(x))$ for all $x \in [0, 1)$. The resulting IRA optimization methods are obtained by using this condition in (25).

While for the inner decoder (accumulator) we are forced to resort to Monte Carlo simulation, it is interesting to notice that, due to the simplicity of the repetition code,

for both Methods 3 and 4 the EXIT function of the outer decoder ($I_E = h(I_A)$) can be obtained in closed form.

For Method 3, by discretizing the channel observation distribution as in (34), we have

$$h(I_A) = 1 - \sum_{i=2}^{d} \sum_{j=1}^{D} \lambda_i p_j \frac{1}{\sqrt{\pi}} \int_{-\infty}^{\infty} e^{-z^2} \log_2 \left(1 + e^{-2\sqrt{(i-1)J^{-1}(I_A)}z - (i-1)J^{-1}(I_A) - v_j}\right) dz \quad (44)$$

For Method 4 we have

$$h(I_A) = 1 - (1 - \mathcal{J}(F_u)) \sum_{i=2}^{d} \lambda_i (1 - I_A)^{i-1} \quad (45)$$

4 Properties of the approximated DE

In this section we show some properties of the approximated DE derived in Section 3.

4.1 Stability condition.

Consider the DE approximation of Method 1. As said in Section 3.2, $(x, \widetilde{x}) = (1, 1)$ is a fixed-point of the system (33–35). We have the following result:

Theorem 2. The fixed-point at $(1, 1)$ of the system (33 – 35) is stable if and only if the fixed-point $(\Delta_\infty, \Delta_\infty)$ of the exact DE (10 – 13) is stable.

Proof. See Appendix A.3. □

For other DE approximations, stability does not generally imply stability of the corresponding exact DE. Consider the DE approximation of Method 2. $(1, 1)$ is a fixed point of the system (41). We have the following result:

Proposition 3. The local stability condition of the approximated DE with Method 2 is less stringent than that of the exact DE.

Proof. See Appendix A.4 □

If an approximated DE has a less stringent stability condition, then the exact stability condition must be added to the ensemble optimization as an explicit additional constraint. It should be noticed that the DE approximations used in [22, 12, 17] require the additional stability constraint. For example, the codes presented in [17] for the BIAWGNC and for which $\lambda_2 > 0$ are not stable. Therefore, the BER for an arbitrary large number of iterations is not vanishing.

4.2 Fixed-points, coding rate and channel capacity.

An interesting property of optimization Methods 2 and 4 is that the optimized ensemble for a given channel with channel observation distribution F_u and capacity $C = \mathcal{I}(F_u)$ has coding rate not larger than C. In fact, as a corollary of a general result of [21] (see Appendix A.5), we have that

Theorem 3. The DE approximations of Methods 2 and 4 have unique fixed-point $(1, 1)$ only if the IRA ensemble coding rate R satisfies $R < C = \mathcal{I}(F_u)$.

Proof. See Appendix A.5 □

We show in Section 5.1 through some examples that this property does not hold in general for other code ensemble optimization methods, for which the ensemble rate R might result to be larger than the (nominal) capacity $\mathcal{I}(F_u)$. This means that the threshold $\nu^\star$, evaluated by exact DE, is worse than the channel parameter ν used for the ensemble design.

5 Numerical results

5.1 Design example for rate 1/2 codes

In this subsection we present the result of optimization for codes of rate 1/2 and give examples for the BSC with cross-over probability p and the BIAWGNC with

$$\text{SNR} \triangleq \frac{E_s}{N_0} = \frac{1}{2\sigma^2}$$

In Fig. 6 the curve is the fixed-point equation used for the optimization in method 1 i.e. the function $\phi(x, \widetilde{x}(x))$. The fixed-point equation curves for the other three methods are very similar.

In Fig. 6 the curve (solid line) shows $\phi(x, \widetilde{x}(x))$ as a function of $x \in [0, 1]$ for method 1. The solutions of the fixed-point equation (23) correspond to the intersection of this curve with the main diagonal (dotted line). Tables 1 and 2 give the degree sequences, the grouping factors and the information bitnode average degrees for the four methods, for codes of rate 1/2 over the BIAWGNC and the BSC, respectively. We compute the true iterative decoding thresholds (by using the exact DE) for all the ensembles (denoted by SNR(DE) or p (DE) in the Tables) and report also the gap of these thresholds with respect to the Shannon limit (denoted by SNR_{gap}(DE) or p (p_{gap}(DE) in the Tables).

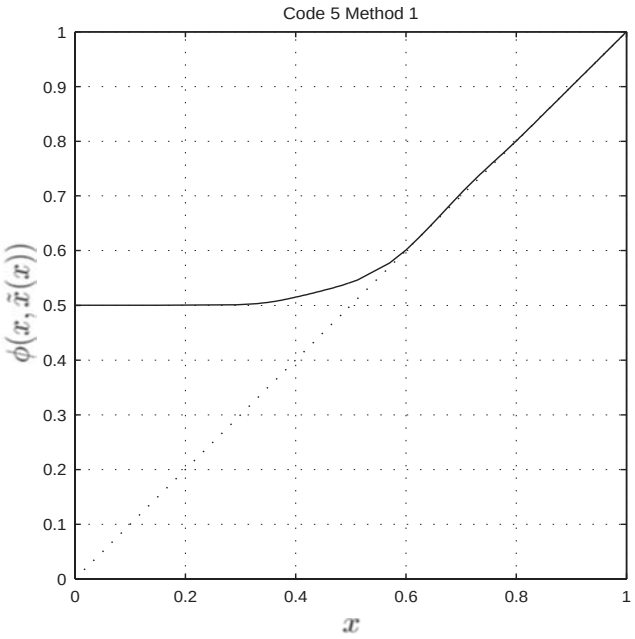

Figure 6: Mutual information EXIT functions for BIAWGNC and Method 1.

Then, we compare it to the threshold of the approximated DE (SNR_{gap}(approx.) and p_{gap}(approx.)). We observe that the codes designed by using methods 2 or 4 have rate below capacity, which is consistent with Theorem 3. On the contrary the codes designed by using methods 1 or 3 have rate possibly larger than the capacity corresponding to the channel parameter used for design. It can easily be checked that all the designed codes are stable.

5.2 Thresholds of IRA ensembles

In this section we present results for codes designed according to the four methods, for rates from 0.1 to 0.9, and we compare the methods on the basis of the true thresholds obtained by DE. We present the code rate, the grouping factor, the average repetition factor and the gap to Shannon limit, for both BSC and BIAWGNC.

Tables 3 and 4 show the performance of IRA codes on the BIAWGNC. Tables 5 and 6 show the performance of IRA codes on the BSC.

For all rates, and for both channels, IRA codes designed assuming GA (Methods 1 and 3) perform much better than those designed assuming BEC a priori (Methods 2 and 4). Nevertheless, Method 4 yields better codes than Method 2, especially at low rates. This is due to the fact that, in Method 2, the communication channel is replaced with a

	Method 1		Method 2		Method 3		Method 4	
	i	λ_i	i	λ_i	i	λ_i	i	λ_i
	2	0.04227	2	0.05554	2	0.05266	2	0.05554
	3	0.16242	3	0.16330	3	0.11786	3	0.14480
	7	0.06529	8	0.06133	5	0.05906	7	0.18991
	8	0.06489	9	0.19357	6	0.06517	8	0.00996
	9	0.06207	25	0.14460	8	0.03615	19	0.03721
	10	0.01273	26	0.08842	9	0.11288	20	0.25894
	11	0.13072	100	0.29323	13	0.06068	100	0.30366
	14	0.04027			14	0.04650		
	25	0.00013			22	0.08606		
	26	0.05410			23	0.01610		
	36	0.13031			34	0.11019		
	37	0.13071			35	0.11919		
	100	0.10402			100	0.11751		
Rate	0.50183		0.49697		0.50154		0.49465	
a	8		8		8		8	
$\bar{d}$	7.94153		8.09755		7.95087		8.17305	
SNR(DE)	-2.739		-2.457		-2.727		-2.588	
SNR_{gap}(DE)	0.059		0.406		0.075		0.306	
SNR_{gap}(approx.)	-0.025		0.040		-0.021		0.071	

Table 1: Optimization for the BIAWGNC

BEC with the same capacity, while this is not the case in Method 4. This difference of performance decreases as the rate increases.

Fig. 7 compares the performance of IRA ensembles with the best known LDPC ensembles [5] on the BIAWGNC. As expected, the performance of IRA ensembles is inferior to that of LDPC ensembles. However, in view of the simplicity of their encoding and decoding, IRA codes, optimized using Methods 1 or 3, emerge as a very attractive design alternative.

Fig. 8 compares the performance of IRA ensembles obtained via the proposed methods for the BSC. The best codes are those designed with Method 3.

6 Conclusions

This paper has tackled the optimization of IRA codes in the limit for large code block-length. This assumption allows to consider a cycle-free graph and enables to evaluate the threshold of the code by iteratively calculating message densities (DE). For the sake of

	Method 1		Method 2		Method 3		Method 4	
	i	λ_i	i	λ_i	i	λ_i	i	λ_i
	2	0.03545	2	0.04732	2	0.03115	2	0.04657
	3	0.14375	3	0.17984	3	0.14991	3	0.14932
	6	0.03057	9	0.19715	6	0.04630	7	0.07693
	7	0.10963	10	0.06259	7	0.06217	8	0.16249
	9	0.10654	26	0.16429	8	0.08666	20	0.07001
	10	0.02388	27	0.05676	10	0.12644	21	0.20550
	11	0.04856	100	0.29205	17	0.03430	100	0.28919
	12	0.00461			18	0.01506		
	21	0.03035			26	0.00228		
	28	0.22576			27	0.02258		
	29	0.09453			28	0.21774		
	100	0.14635			29	0.08021		
					100	0.12521		
Rate	0.48908		0.49620		0.49226		0.49091	
a	8		8		8		8	
$\bar{d}$	8.35724		8.12253		8.25157		8.29627	
p(DE)	0.1091		0.0938		0.1091		0.1009	
p_{gap}(DE)	0.0046		0.0175		0.0035		0.0122	
p_{gap}(approx.)	0.0037		0.0013		0.0026		0.0018	

Table 2: Optimization for the BSC

tractable analysis, we proposed four methods to approximate those densities as a one-dimensional parameter. These approximations were motivated by recent results in the field of code design (EXIT functions, reciprocal channel approximation, and the non-strict convexity of mutual information) and have led to four optimization methods that can all be solved as a linear program.

We found a general stability condition for IRA codes under exact DE. We showed formally that one of the proposed methods (Gaussian approximation, with reciprocal channel approximation) yields a one-dimensional DE approximation with the same stability condition, whereas the exact stability condition must be added to the ensemble optimization as an explicit additional constraint for another method (BEC a priori, with reciprocal channel approximation). We derived also results related to the rates of the codes: in general the Gaussian a priori methods are optimistic, in the sense that there is no guarantee that the optimized rate is below capacity. On the contrary, the BEC a priori methods have always rates below capacity.

Our numerical results show that, for the BIAWGNC and BSC, the Gaussian a priori

Method 1				Method 3			
Rate	a	$\overline{d}$	SNR_{gap}	Rate	a	$\overline{d}$	SNR_{gap}
0.10109	2	17.78	0.151	0.10133	2	17.74	0.163
0.20191	3	11.86	0.096	0.20199	3	11.85	0.126
0.30153	4	9.27	0.081	0.30175	4	9.26	0.111
0.40196	6	8.93	0.057	0.40401	6	8.85	0.067
0.50184	8	7.94	0.059	0.50154	8	7.95	0.075
0.60188	11	7.28	0.065	0.60147	11	7.29	0.065
0.70154	16	6.81	0.067	0.70093	16	6.83	0.068
0.79904	29	7.29	0.066	0.79912	29	7.29	0.062
0.89677	61	7.02	0.088	0.89712	61	7.00	0.083

Table 3: IRA codes, designed with Methods 1 and 3, for BIAWGNC

Method 2				Method 4			
Rate	a	$\overline{d}$	SNR_{gap}	Rate	a	$\overline{d}$	SNR_{gap}
0.09407	2	19.26	0.906	0.09752	2	18.51	0.316
0.19842	3	12.12	0.573	0.19725	3	12.21	0.293
0.29767	4	9.44	0.529	0.29671	4	9.48	0.336
0.39703	6	9.11	0.466	0.39445	6	9.21	0.343
0.49697	8	8.10	0.406	0.49465	8	8.17	0.306
0.59689	11	7.43	0.362	0.59577	11	7.46	0.338
0.69580	16	7.00	0.323	0.69584	16	6.99	0.296
0.79737	26	6.61	0.272	0.79678	26	6.63	0.271
0.89827	56	6.34	0.212	0.89826	56	6.34	0.214

Table 4: IRA codes, designed with Methods 2 and 4, for BIAWGNC

approximation is more attractive since the codes designed under this assumption have the smallest gap to Shannon limit. Depending on the desired rate, the EXIT function of the inner decoder has to be computed either with Monte-Carlo simulation (Method 3) or with the reciprocal channel approximation (Method 1). At least in the BIAWGNC there is some evidence that the best LDPC codes [5] slightly outperform our designed codes. However, the performance-complexity tradeoff of the optimized IRA codes is quite impressive.

APPENDIX

Method 1				Method 3			
Rate	a	$\overline{d}$	p_{gap}	Rate	a	$\overline{d}$	p_{gap}
0.10042	2	17.92	0.0032	0.10137	2	17.73	0.0036
0.19910	3	12.07	0.0037	0.20086	3	11.94	0.0041
0.29573	4	9.53	0.0044	0.29897	4	9.38	0.0031
0.39298	6	9.27	0.0044	0.39621	6	9.14	0.0032
0.48908	8	8.36	0.0046	0.49226	8	8.25	0.0035
0.58590	12	8.48	0.0044	0.58815	12	8.40	0.0040
0.68271	17	7.90	0.0044	0.68409	16	7.39	0.0039
0.78155	28	7.83	0.0038	0.78235	28	7.79	0.0035
0.88437	59	7.71	0.0026	0.88457	63	8.22	0.0025

Table 5: IRA codes, designed with Methods 1 and 3, for BSC.

Method 2				Method 4			
Rate	a	$\overline{d}$	p_{gap}	Rate	a	$\overline{d}$	p_{gap}
0.09406	2	19.26	0.0194	0.09952	2	18.10	0.0121
0.19833	3	12.13	0.0175	0.19842	3	12.12	0.0101
0.29743	4	9.45	0.0190	0.28836	4	9.87	0.0114
0.39650	6	9.13	0.0187	0.38865	6	9.44	0.0149
0.49620	8	8.12	0.0175	0.49091	8	8.30	0.0122
0.59580	11	7.46	0.0155	0.59349	11	7.53	0.0124
0.69559	16	7.00	0.0126	0.69107	16	7.15	0.0116
0.79583	26	6.67	0.0091	0.79283	26	6.79	0.0090
0.89692	57	6.55	0.0049	0.89337	57	6.80	0.0051

Table 6: IRA codes, designed with Methods 2 and 4, for BSC.

A Proofs

A.1 Proof of Theorem 1

We follow in the footsteps of [9] and analyze the local stability of the zero-BER fixed-point by using a small perturbation approach. In order to do this, we need more details on the mapping Γ and its inverse.

Given a random variable x with distribution $F_x(z)$, the distribution of $\gamma(x)$ is given by:

$$\Gamma(F_x)(s,z) = \chi_{\{s=0\}}\Gamma_0(F_x)(z) + \chi_{\{s=1\}}\Gamma_1(F_x)(z) \tag{46}$$

where

$$\Gamma_0(F_x)(z) = 1 - F_x^-(-\ln\tanh\frac{z}{2}),$$

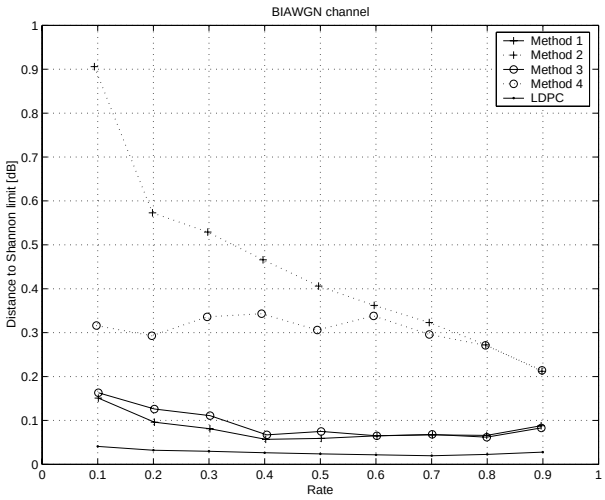

Figure 7: Gap to Shannon limit vs. rate for BIAWGNC.

$$\Gamma_1(F_x)(z) = F_x(\ln\tanh\frac{z}{2}),$$

and where $\chi_{\mathcal{A}}$ denotes the indicator function of the event $\mathcal{A}$.

In particular, the mapping Γ applied to Δ_0 and Δ_∞ yields

$$\begin{aligned}\Gamma(\Delta_0)(s,z) &= \frac{1}{2}\chi_{\{s=0\}}\Delta_\infty(z) + \frac{1}{2}\chi_{\{s=1\}}\Delta_\infty(z)\\ \Gamma(\Delta_\infty)(s,z) &= \chi_{\{s=0\}}\Delta_0(z).\end{aligned} \tag{47}$$

Given $G(s,z) = \chi_{\{s=0\}}G_0(z) + \chi_{\{s=1\}}G_1(z)$, applying Γ^{-1} yields

$$\Gamma^{-1}(G)(z) = \chi_{\{z>0\}}(1 - G_0(-\log\tanh\frac{z}{2})) + \chi_{\{z<0\}}G_1(-\log\tanh\frac{-z}{2}) \tag{48}$$

For the sake of brevity, we introduce the short-hand notation

$$G(s,z) = \chi_{\{s=0\}}G_0(z) + \chi_{\{s=1\}}G_1(z) = \chi_0 G_0 + \chi_1 G_1$$

The m-fold convolution of $G(s,z)$ by itself is given by

$$(\chi_0 G_0(z)+\chi_1 G_1(z))^{\otimes m} = \chi_0\left(\sum_{j\ even, j=0}^{m}\binom{m}{j}G_0^{\otimes(m-j)}\otimes G_1^{\otimes j}\right)+\chi_1\left(\sum_{j\ odd, j=1}^{m}\binom{m}{j}G_0^{\otimes(m-j)}\otimes G_1^{\otimes j}\right) \tag{49}$$

In order to study the local stability of the fixed-point $(\Delta_\infty, \Delta_\infty)$, we initialize the DE recursion at the point

$$\begin{cases} P_0 &= (1-2\epsilon)\Delta_\infty + 2\epsilon\Delta_0\\ \widetilde{P}_0 &= (1-2\delta)\Delta_\infty + 2\delta\Delta_0\end{cases}$$

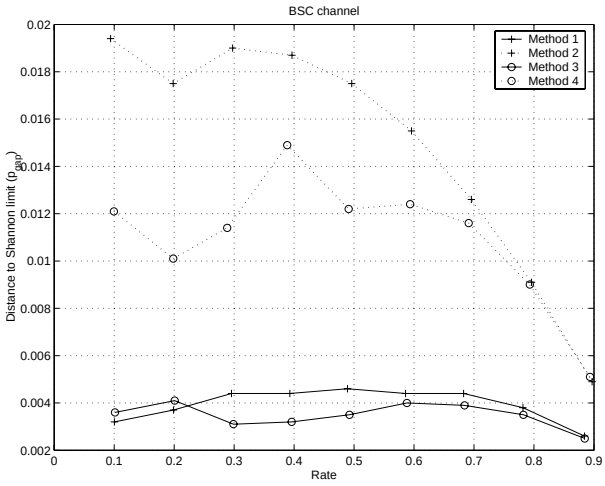

Figure 8: Gap to Shannon limit vs. rate for BSC.

for some small $\epsilon, \delta > 0$, and we apply one iteration of the DE recursion (10 – 13). The step-by-step derivation is as follows. From (47) we have

$$\begin{cases} \Gamma(P_0) &= \chi_0\left((1-2\epsilon)\Delta_0 + \epsilon\Delta_\infty\right) + \chi_1\left(\epsilon\Delta_\infty\right) \\ \Gamma(\widetilde{P}_0) &= \chi_0\left((1-2\delta)\Delta_0 + \delta\Delta_\infty\right) + \chi_1\left(\delta\Delta_\infty\right) \end{cases}$$

By applying (49) we obtain

$$\begin{cases} \Gamma(P_0)^{\otimes n} &= \chi_0\left((1-2n\epsilon)\Delta_0 + n\epsilon\Delta_\infty\right) + \chi_1\left(n\epsilon\Delta_\infty\right) + O(\epsilon^2) \\ \Gamma(\widetilde{P}_0)^{\otimes 2} &= \chi_0\left((1-4\delta)\Delta_0 + 2\delta\Delta_\infty\right) + \chi_1\left(2\delta\Delta_\infty\right) + O(\delta^2) \end{cases}$$

By applying Γ^{-1} we get

$$\begin{cases} Q_1 = \Gamma^{-1}\left(\Gamma(P_0)^{\otimes(a-1)} \otimes \Gamma(\widetilde{P}_0)^{\otimes 2}\right) &= (1-2(a-1)\epsilon - 4\delta)\Delta_\infty + (2(a-1)\epsilon + 4\delta)\Delta_0 + O(\epsilon^2, \delta^2) \\ \widetilde{Q}_1 = \Gamma^{-1}\left(\Gamma(P_0)^{\otimes a} \otimes \Gamma(\widetilde{P}_0)\right) &= (1-2a\epsilon - 2\delta)\Delta_\infty + (2a\epsilon + 2\delta)\Delta_0 + O(\epsilon^2, \delta^2) \end{cases}$$

Hence, by noticing that

$$\begin{aligned} Q_1^{\otimes n} &= \sum_{j=0}^{n} \binom{n}{j} (1-2(a-1)\epsilon - 4\delta)^{n-j} (2(a-1)\epsilon + 4\delta)^j \Delta_\infty^{\otimes n-j} \otimes \Delta_0^{\otimes j} + O(\epsilon^2, \delta^2) \\ &= \begin{cases} \Delta_\infty + O(\epsilon^2, \delta^2), & \text{for } n \geq 2 \\ (1-2(a-1)\epsilon - 4\delta)\Delta_\infty + (2(a-1)\epsilon + 4\delta)\Delta_0 + O(\epsilon^2, \delta^2), & \text{for } n = 1 \end{cases} \end{aligned}$$

we have

$$\lambda(Q_1) = (1-2(a-1)\lambda_2\epsilon - 4\lambda_2\delta)\Delta_\infty + (2(a-1)\lambda_2\epsilon + 4\lambda_2\delta)\Delta_0 + O(\epsilon^2, \delta^2).$$

Finally, by using the fact that $P_1 = F_u \otimes \lambda(Q_1)$ and that $\widetilde{P}_1 = F_u \otimes \widetilde{Q}_1$, the message distributions after one DE iteration are given by

$$\begin{bmatrix} P_1 \\ \widetilde{P}_1 \end{bmatrix} = \mathbf{A} \begin{bmatrix} 2\epsilon \\ 2\delta \end{bmatrix} F_u + \left(\begin{bmatrix} 1 \\ 1 \end{bmatrix} - \mathbf{A} \begin{bmatrix} 2\epsilon \\ 2\delta \end{bmatrix} \right) \Delta_\infty + \begin{bmatrix} O(\epsilon^2) \\ O(\delta^2) \end{bmatrix}$$

where

$$\mathbf{A} = \begin{bmatrix} (a-1)\lambda_2 & 2\lambda_2 \\ a & 1 \end{bmatrix} \tag{50}$$

After ℓ iterations we obtain

$$\begin{bmatrix} P_\ell \\ \widetilde{P}_\ell \end{bmatrix} = \mathbf{A}^\ell \begin{bmatrix} 2\epsilon \\ 2\delta \end{bmatrix} F_u^{\otimes \ell} + \left(\begin{bmatrix} 1 \\ 1 \end{bmatrix} - \mathbf{A}^\ell \begin{bmatrix} 2\epsilon \\ 2\delta \end{bmatrix} \right) \Delta_\infty + \begin{bmatrix} O(\epsilon^2) \\ O(\delta^2) \end{bmatrix} \tag{51}$$

From the large deviation theory we get that [9]

$$\begin{aligned} r &= -\lim_{\ell \to \infty} \frac{1}{\ell} \log \mathrm{Pe}(F_u^{\otimes \ell}) \\ &= -\log \left(\inf_{s>0} \int e^{-sz} dF_u(z) \right) \\ &= -\log \left(\int e^{-z/2} dF_u(z) \right) \end{aligned} \tag{52}$$

where the last equality follows from the fact that $F_u(z) \in \mathcal{F}_{\mathrm{sym}}$.

Then, by applying $\mathrm{Pe}(\cdot)$ to P_ℓ in (51) we obtain that $\lim_{\ell \to \infty} \mathrm{Pe}(P_\ell) = 0$ (implying that $\lim_{\ell \to \infty} P_\ell = \Delta_\infty$) if the eigenvalues of the matrix $\mathbf{A}e^{-r}$ are inside the unit circle.

The stability condition is obtained by computing explicitly the largest (in magnitude) eigenvalue. We obtain

$$\frac{1}{2} \left(1 + \lambda_2(a-1) + \sqrt{1 + (2+6a)\lambda_2 + (a-1)^2 \lambda_2^2} \right) < e^r. \tag{53}$$

Since the LHS of (53) is increasing, condition (53) is indeed an upperbound on λ_2, given explicitly by (15).

A.2 Proof of Proposition 2

Let S be a discrete random variable taking on the values $\{1, \ldots, m\}$ with probabilities $q_1, \ldots, q_m$. Let $\{F_1, \ldots, F_m\}$ be a collection of symmetric distributions, and let

$$F(y) = \sum_{i=1}^{m} q_i F_i(y)$$

Then, define the collection of binary-input symmetric-output channels $\{p_{Y|X,S} : s = 1, \ldots, m\}$ such that

$$p_{Y|X,S}(y|0, s = i) = F_i(y)$$

and where S and X are independent. Let $X \sim (p, 1-p)$. For simplicity, we assume that the distributions $\{F_i\}$ have densities $\{f_i\}$.

The assertion of Proposition 2 is proved by showing that $I(X;Y) = I(X;Y|S)$:

$$\begin{aligned}
I(X;Y) &= p \int \log_2 \frac{f(y)}{pf(y) + (1-p)f(-y)} f(y)dy \\
&+ (1-p) \int \log_2 \frac{f(-y)}{pf(y) + (1-p)f(-y)} f(-y)dy \\
&= p \int \log_2 \frac{1}{p + (1-p)e^{-y}} f(y)dy + (1-p) \int \log_2 \frac{1}{pe^{-y} + (1-p)} f(y)dy \\
&= \int \left(p \log_2 \frac{1}{p + (1-p)e^{-y}} + (1-p) \int \log_2 \frac{1}{pe^{-y} + (1-p)} \right) \sum_{i=1}^{m} q_i f_i(y)dy \\
&= \sum_{i=1}^{m} q_i \int \left(p \log_2 \frac{1}{p + (1-p)e^{-y}} + (1-p) \int \log_2 \frac{1}{pe^{-y} + (1-p)} \right) f_i(y)dy \\
&= \sum_{i=1}^{m} q_i \left(p \int \log_2 \frac{f_i(y)}{pf_i(y) + (1-p)f_i(-y)} f_i(y)dy \right. \\
&\qquad \left. + (1-p) \int \log_2 \frac{f_i(-y)}{pf_i(y) + (1-p)f_i(-y)} f_i(-y)dy \right) \\
&= I(X;Y|S) \qquad (54)
\end{aligned}$$

A.3 Proof of Theorem 2

The local stability condition for the system ((33) and (35)) is given by the eigenvalues of the Jacobian matrix for the functions $(\phi, \tilde{\phi})$ in the fixed point $(x, \tilde{x}) = (1, 1)$. The partial derivatives of ϕ and $\tilde{\phi}$ are

$$\frac{\partial \phi}{\partial x}(1,1) = \sum_{i=2}^{d}\sum_{j=1}^{D} \lambda_i p_j (i-1)(a-1) \lim_{\mu\to+\infty} \frac{J'_{v_j}((i-1)\mu)}{J'(\mu)} \quad (55)$$

$$\frac{\partial \phi}{\partial \tilde{x}}(1,1) = \sum_{i=2}^{d}\sum_{j=1}^{D} \lambda_i p_j (i-1) 2 \lim_{\mu\to+\infty} \frac{J'_{v_j}((i-1)\mu)}{J'(\mu)} \quad (56)$$

$$\frac{\partial \tilde{\phi}}{\partial x}(1,1) = \sum_{j=1}^{D} p_j a \lim_{\mu\to+\infty} \frac{J'_{v_j}(\mu)}{J'(\mu)} \quad (57)$$

$$\frac{\partial \tilde{\phi}}{\partial \tilde{x}}(1,1) = \sum_{j=1}^{D} p_j a \lim_{\mu\to+\infty} \frac{J'_{v_j}(\mu)}{J'(\mu)} \quad (58)$$

where

$$J_{v_j}(\mu) \triangleq 1 - \frac{1}{\sqrt{\pi}} \int_{-\infty}^{+\infty} e^{-z^2} \log\left(1 + e^{-2\sqrt{\mu}z - \mu - v_j}\right) dz. \quad (59)$$

Note that $J_0(\mu) = J(\mu)$. Since both limits tend to 0, we derive an asymptotic expansion for $J'_{v_j}(\mu)$ and $J'(\mu)$.

The derivative of J_{v_j} is given by

$$J'_{v_j}(\mu) = \frac{\log(e)}{\sqrt{\mu}} \frac{1}{\sqrt{\pi}} \int_{-\infty}^{+\infty} (z + \sqrt{\mu}) e^{-v_j} \frac{e^{-(z+\sqrt{\mu})^2}}{1 + e^{-2\sqrt{\mu}z - \mu - v_j}} dz$$

Since F_u is symmetric, the sum over j can be rewritten as:

$$\sum_{j=1}^{D} p_j J'_{v_j}(\mu) = p_0 J'_0(\mu) + \sum_{j=1}^{D'} p_j \left(J'_{v_j}(\mu) + e^{-v_j} J'_{-v_j}(\mu)\right)$$

Let us define

$$f_0(\mu) = \frac{1}{\log(e)} J'_0(\mu)$$

$$f_{v_j}(\mu) = \frac{1}{\log(e)} \left(J'_{v_j}(\mu) + e^{-v_j} J'_{-v_j}(\mu)\right)$$

$$= \frac{1}{\sqrt{\pi}} \int_{-\infty}^{+\infty} (1 + \frac{z}{\sqrt{\mu}}) e^{-(z+\sqrt{\mu})^2} \left(\frac{e^{-v_j}}{1 + e^{-2\sqrt{\mu}z - \mu - v_j}} + \frac{1}{1 + e^{-2\sqrt{\mu}z - \mu + v_j}}\right) dz \quad (60)$$

Following [34], (60) can be rewritten as

$$\begin{aligned}
f_{v_j}(\mu) &= \frac{1}{\sqrt{\pi}}\int_{-\infty}^{+\infty}\frac{1}{\sqrt{\mu}}\left(z+\frac{\sqrt{\mu}}{2}\right)e^{-\left(z+\frac{\sqrt{\mu}}{2}\right)^2}\left(\frac{e^{-v_j}}{1+e^{-2\sqrt{\mu}z-v_j}}+\frac{1}{1+e^{-2\sqrt{\mu}z+v_j}}\right)dz \\
&= \frac{1}{\sqrt{\pi}}\int_{-\infty}^{+\infty}\frac{z}{\sqrt{\mu}}e^{-z^2-\frac{\mu}{4}-\frac{v_j}{2}}\left(\frac{1}{e^{\sqrt{\mu}z+\frac{v_j}{2}}+e^{-\sqrt{\mu}z-\frac{v_j}{2}}}+\frac{1}{e^{\sqrt{\mu}z-\frac{v_j}{2}}+e^{-\sqrt{\mu}z+\frac{v_j}{2}}}\right)dz \\
&\quad +\frac{1}{\sqrt{\pi}}\int_{-\infty}^{+\infty}\frac{1}{2}e^{-z^2-\frac{\mu}{4}-\frac{v_j}{2}}\left(\frac{1}{e^{\sqrt{\mu}z+\frac{v_j}{2}}+e^{-\sqrt{\mu}z-\frac{v_j}{2}}}+\frac{1}{e^{\sqrt{\mu}z-\frac{v_j}{2}}+e^{-\sqrt{\mu}z+\frac{v_j}{2}}}\right)dz \\
&= \frac{e^{-\frac{\mu}{4}-\frac{v_j}{2}}}{4\sqrt{\pi}}\int_{-\infty}^{+\infty}e^{-z^2}\left(\frac{1}{ch(\sqrt{\mu}z+\frac{v_j}{2})}+\frac{1}{ch(\sqrt{\mu}z-\frac{v_j}{2})}\right)dz \\
&= \frac{e^{-\frac{\mu}{4}-\frac{v_j}{2}}}{4\sqrt{\pi\mu}}\int_{-\infty}^{+\infty}\frac{e^{-\frac{(z-\frac{v_j}{2})^2}{\mu}}+e^{-\frac{(z+\frac{v_j}{2})^2}{\mu}}}{ch(z)}dz \qquad (61)
\end{aligned}$$

The first equality in (61) is obtained by the change of variable $z' = z + \sqrt{\mu}/2$. The third equality is due to the fact that the first and second integrands in the second line of (61) are odd and even functions of z, respectively. Then we use the changes of variable $z' = \sqrt{\mu}z + \frac{v_j}{2}$ and $z' = \sqrt{\mu}z - \frac{v_j}{2}$.

Lebesgue's dominated convergence theorem completes the proof. Since the sequence of measurable functions verifies:

$$\forall z \in \mathbb{R}, \frac{e^{-\frac{z^2}{\mu}}}{ch(z)} \xrightarrow[\mu\to+\infty]{} \frac{1}{ch(z)}$$

and since these functions are bounded by an integrable function independent of μ:

$$\forall \mu > 0, \forall z \in \mathbb{R}, \left|\frac{e^{-\frac{z^2}{\mu}}}{ch(z)}\right| \leq \frac{1}{ch(z)} \in L^1(\mathbb{R}).$$

Thus Lebesgue's dominated convergence theorem applies and

$$\int_{-\infty}^{+\infty}\frac{e^{-\frac{z^2}{\mu}}}{ch(z)}dz \xrightarrow[\mu\to+\infty]{} \int_{-\infty}^{+\infty}\frac{1}{ch(z)}dz = [2\arctan(e^z)]_{-\infty}^{+\infty} = \pi$$

Therefore for large μ

$$f_{v_j}(\mu) \sim \frac{\sqrt{\pi}}{2}\,\frac{e^{-\frac{\mu}{4}}e^{-\frac{v_j}{2}}}{\sqrt{\mu}}$$

Similarly we get

$$f_0(\mu) \sim \frac{\sqrt{\pi}}{4}\,\frac{e^{-\frac{\mu}{4}}}{\sqrt{\mu}}$$

And thus, for $n \geq 1$

$$\lim_{\mu \to +\infty} \frac{f_{v_j}(n\mu)}{f_0(\mu)} = \begin{cases} 2e^{-\frac{v_j}{2}} & \text{if} \quad n = 1 \\ 0 & \text{if} \quad n > 1 \end{cases}$$

and

$$\lim_{\mu \to +\infty} \frac{f_0(n\mu)}{f_0(\mu)} = \begin{cases} 1 & \text{if} \quad n = 1 \\ 0 & \text{if} \quad n > 1 \end{cases}$$

The partial derivatives of ϕ and $\tilde{\phi}$ are

$$\begin{aligned} \frac{\partial \phi}{\partial x}(1,1) &= \lambda_2(a-1)(p_0 + \sum_{j=1}^{D'} 2p_j e^{-\frac{v_j}{2}}) \\ &= \lambda_2(a-1)\sum_{j=1}^{D} p_j e^{-\frac{v_j}{2}} \\ &= \lambda_2(a-1)e^{-r} \end{aligned} \tag{62}$$

where r is defined in (52). Similarly,

$$\frac{\partial \phi}{\partial \tilde{x}}(1,1) = \lambda_2 2e^{-r} \tag{63}$$

$$\frac{\partial \tilde{\phi}}{\partial x}(1,1) = ae^{-r} \tag{64}$$

$$\frac{\partial \tilde{\phi}}{\partial \tilde{x}}(1,1) = e^{-r} \tag{65}$$

We get the Jacobian matrix as:

$$\mathbf{J} = \begin{bmatrix} (a-1)\lambda'(0) & 2\lambda'(0) \\ a & 1 \end{bmatrix} e^{-r}$$

In order to be stable the eigenvalues of $\mathbf{J}$ should be inside the unit circle. Therefore the stability condition reduces to:

$$\frac{1}{2}\left(1 + \lambda'(0)(a-1) + \sqrt{1 + 2\lambda'(0) + 6\lambda'(0)a + \lambda'(0)^2(a-1)^2}\right) < e^{-r}. \tag{66}$$

Notice from (53) and (66) that the stability conditions under DE and approximated DE are the same.

A.4 Proof of proposition 3

The Jacobian matrix of the approximated DE (41) about the fixed point $[x, \tilde{x}] = [1, 1]$, for a given input channel distribution F_u, is

$$\mathbf{J} = \begin{bmatrix} (a-1)\lambda'(0) & 2\lambda'(0) \\ a & 1 \end{bmatrix} (1 - \mathcal{I}(F_u)) = \mathbf{A}(1 - \mathcal{I}(F_u))$$

where $\mathbf{A}$ was already defined in (50). The stability of the exact DE is given by the eigenvalues of $\mathbf{A}e^{-r}$ (where r is defined in (52)) while it is given by those of $\mathbf{A}(1-\mathcal{I}(F_u))$ for the approximated DE. From the inequality

$$\forall z \in \mathbb{R} \qquad \log_2(1 + \mathrm{e}^{-z}) \leq \mathrm{e}^{-z/2},$$

we obtain for all distribution F_u

$$\int \log_2(1 + e^{-z}) dF_u(z) \leq \int e^{-z/2} dF_u(z).$$

Under the assumption that F_u is symmetric and from the definition of $\mathcal{I}(F)$ given in (28), we get

$$\forall F_u \in \mathcal{F}_{sym} \qquad 1 - \mathcal{I}(F_u) \leq e^{-r}$$

and the conclusion follows.

A.5 Proof of Theorem 3

Theorem 3 follows as a corollary of a result of [21] that we state here for the sake of completeness as Lemma 2 below. In order to introduce this result, we consider the model of Fig. 9, where $\mathbf{b}$, $\mathbf{x}_1$ and $\mathbf{x}$ are binary sequences and where Channel 1 is the communication channel with output $\mathbf{y}$ and Channel 2 is a BEC channel with output $\mathbf{z}$. Let the decoder be a MAP symbol-by-symbol decoder, producing for all $i = 1, \ldots, n$, output messages of the form

$$m_{o,i} = \log \frac{P(x_{1,i} = 0 | \mathbf{y}, \mathbf{z}_{[i]})}{P(x_{1,i} = 1 | \mathbf{y}, \mathbf{z}_{[i]})} \tag{67}$$

where $\mathbf{z}_{[i]} \stackrel{\Delta}{=} (z_1, \ldots, z_{i-1}, z_{i+1}, \ldots, z_n)$. Following [21], we generalize the definition of I_A

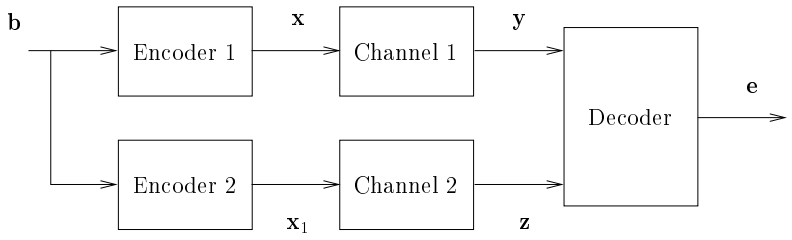

Figure 9: General decoding model.

and I_E given in Section (3.1) to the case of sequences as

$$\begin{aligned} I_A &= \frac{1}{n}\sum_{i=1}^{n} I(x_{1,i}; z_i) \\ I_E &= \frac{1}{n}\sum_{i=1}^{n} I(x_{1,i}; m_{o,i}) \\ &\stackrel{\text{a}}{=} \frac{1}{n}\sum_{i=1}^{n} I(x_{1,i}; \mathbf{y}, \mathbf{z}_{[i]}) \end{aligned} \tag{68}$$

where (a) follows from the fact that the decoder is MAP. Again, the decoder EXIT function is the set of points (I_A, I_E) for all $I_A \in [0, 1]$.

For the setup of Fig. 9 with the above assumptions, the following result applies:

Lemma 2 [21] Let $\mathbf{b}$ be uniformly distributed and i.i.d.. If Encoder 2 is linear with generator matrix having no all-zero columns, then the area under the EXIT characteristic satisfies

$$\mathcal{A} \stackrel{\Delta}{=} \int_0^1 I_E(z)dz = 1 - \frac{1}{n}H(\mathbf{x}_1|\mathbf{y}) \tag{69}$$

□

We start by proving Theorem 3 for the approximated DE of Method 4. Consider the IRA encoder of Fig. 1 and the turbo-like decoder of Fig. 5.

The inner MAP decoder receives channel observations $\mathbf{u}_p$ for the parity bits and input messages for the symbols of $\mathbf{x}_1$, and produces output messages for the symbols of $\mathbf{x}_1$. The general decoding model of Fig. 9, applied to the inner decoder, yields the model of Fig. 10 (a).

The outer MAP decoder receives channel observations $\mathbf{u}_s$ for the information bits and input messages for the symbols of $\mathbf{x}_1$, and produces output messages for the symbols of $\mathbf{x}_1$. The general decoding model of Fig. 9, applied to the outer decoder, yields the model of Fig. 10 (b).

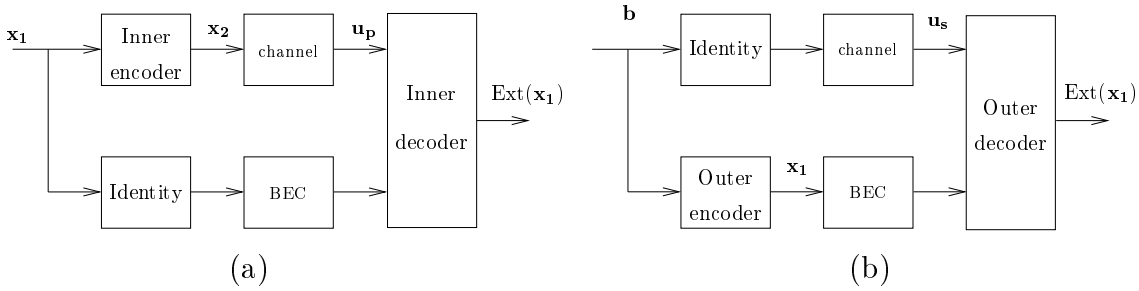

Figure 10: Model of inner and outer decoders Method 4

The upper channel is the communication channel with capacity $\mathcal{I}(F_u)$. Since we consider approximation Method 4, we let lower channel to be a BEC in both Figs. 10 (a) and (b). Let k, n and m denote the number of information bits (length of $\mathbf{b}$ and of $\mathbf{u}_s$), the number of repeated information bits (length of $\mathbf{x}_1$) and the number of parity bits (length of $\mathbf{x}_2$ and of $\mathbf{u}_p$), respectively. The inner and outer coding rates are $R_{\text{in}} = n/m$ and $R_{\text{out}} = k/n$, and the overall IRA coding rate (3) is given by

$$R = \frac{k}{k+m} = \frac{R_{\text{in}} R_{\text{out}}}{1 + R_{\text{in}} R_{\text{out}}}$$

By applying Lemma 2 to the inner code model (Fig. 10 (a)), we obtain

$$\begin{aligned} \mathcal{A}_{\text{in}} &= 1 - \frac{1}{n} H(\mathbf{x}_1 | \mathbf{u}_p) \\ &= 1 - \frac{1}{n} (H(\mathbf{x}_1) - I(\mathbf{x}_1; \mathbf{u}_p)) \\ &\stackrel{\text{a}}{=} \frac{1}{n} I(\mathbf{x}_1; \mathbf{u}_p) \\ &\stackrel{\text{b}}{=} \frac{m}{n} I(x_{2,i}; u_{p,i}) = \mathcal{I}(F_u)/R_{\text{in}} \end{aligned} \tag{70}$$

where (a) follows from the fact that, by the model assumption, $\mathbf{x}_1$ is an i.i.d. uniformly distributed binary sequence, and (b) follows from the fact that the accumulator (inner code) generates $\mathbf{x}_2$ with uniform probability (and uniform marginals) if driven by the i.i.d. uniform input sequence $\mathbf{x}_1$.

By applying Lemma 2 to the outer code model ((Fig. 10 (b)), we obtain

$$\begin{aligned} \mathcal{A}_{\text{out}} &= 1 - \frac{1}{n} H(\mathbf{x}_1 | \mathbf{u}_s) \\ &= 1 - \frac{1}{n} (H(\mathbf{x}_1) - I(\mathbf{x}_1; \mathbf{u}_s)) \\ &\stackrel{\text{a}}{=} 1 - \frac{k}{n} + \frac{1}{n} I(\mathbf{x}_1; \mathbf{u}_s) \\ &\stackrel{\text{b}}{=} 1 - \frac{k}{n} + \frac{k}{n} I(b_i; u_{s,i}) = 1 - R_{\text{out}} + R_{\text{out}} \mathcal{I}(F_u) \end{aligned} \tag{71}$$

where both (a) and (b) follow from the fact that the repetition code is an invertible mapping, so the entropy $H(\mathbf{x}_1)$ is equal to the entropy of the information sequence $\mathbf{b}$ (equal to k bits) and $I(\mathbf{x}_1;\mathbf{u}_s) = I(\mathbf{b};\mathbf{u}_s) = kI(b_i;u_{s,i}) = k\mathcal{I}(F_u)$.

As seen in Section 3.4, the approximated DE has no fixed-points other than $(1,1)$ if and only if $g(x) > h^{-1}(x)$ for all $x \in [0,1)$, where $g(x)$ and $h(x)$ denote the inner and outer decoder EXIT functions. This implies that

$$\mathcal{A}_{\text{in}} = \int_0^1 g(x)dx > \int_0^1 h^{-1}(x)dx = 1 - \mathcal{A}_{\text{out}}$$

By using (70) and (71), we obtain

$$\begin{aligned} \mathcal{I}(F_u)/R_{\text{in}} &> R_{\text{out}} - R_{\text{out}}\mathcal{I}(F_u) \\ &\Downarrow \\ \mathcal{I}(F_u) &> \frac{R_{\text{in}}R_{\text{out}}}{1 + R_{\text{in}}R_{rmout}} = R \end{aligned} \tag{72}$$

For Method 2, the above derivation still holds, since the communication channel in Fig.9 is replaced by a BEC with erasure probablity $\epsilon = 1 - \mathcal{I}(F_u)$. In fact, the inner and outer decoder EXIT functions can be rewritten as

$$\begin{aligned} h(x) &= 1 - (1 - \mathcal{I}(F_u)) \sum_{i=2}^{d} \lambda_i \, (1-x)^{i-1} \\ g(x) &= \frac{x^{a-1}\mathcal{I}(F_u)^2}{(1 - (1 - \mathcal{I}(F_u))x^a)^2} \end{aligned}$$

and the area under these functions are again

$$\begin{aligned} \mathcal{A}_{\text{out}} &= \int_0^1 h(x)dx = 1 - (1 - \mathcal{I}(F_u)) \sum_{i=2}^{d} \lambda_i / i = 1 - R_{\text{out}} + R_{\text{out}}\mathcal{I}(F_u) \\ \mathcal{A}_{\text{in}} &= \int_0^1 g(x)dx = \mathcal{I}(F_u)/a = \mathcal{I}(F_u)/R_{\text{in}} \end{aligned}$$

Therefore, the final result (72) holds also for Method 2.

Acknowledgement

The authors wish to thank Dr. Alex Ashikhmin for the helpful discussion concerning the results in [21].

References

[1] C. Berrou, A. Glavieux, P. Thitimajshima, "Near Shannon limit error-correcting and decoding: Turbo codes," *Proc. ICC*, Geneva, pp. 1064-1070, May 1993.

[2] R.G. Gallager, *Low-Density Parity-Check Codes*, MIT Press, Cambridge, MA, 1963.

[3] M. Luby, M. Mitzenmacher, A. Shokrollahi, and D. Spielman, "Analysis of low-density codes and improved designs using irregular graphs," in *Proceedings of the 30th ACM Symposium on Theory of Computing*, 1998, pp. 249-258.

[4] D. Divsalar, H. Jin, and R. McEliece, "Coding theorems for 'Turbo-like' codes," Proceedings of *the 36th Annual Allerton Conference on Communication, Control, and Computing*, Sept 1998.

[5] R. Urbanke et al., "Web page," http://lthcwww.epfl.ch/research/ldpcopt/,2002.

[6] N. Varnica, A. Kavcic, "Optimized LDPC codes for partial response channels," in *Proc. IEEE Int. Symposium Information Theory (ISIT 2002)*, Lausanne, Switzerland, July 2002, p. 197.

[7] Xiao Ma, N. Varnica, A. Kavcic, "Matched information rate codes for binary ISI channels," in *Proc. IEEE Int. Symposium Information Theory (ISIT 2002)*, Lausanne, Switzerland, July 2002, p. 269.

[8] B.M. Kurkoski, P.H. Siegel, J.K. Wolf, "Joint message-passing decoding of LDPC codes and partial-response channels," *IEEE Trans. on Inf. Theory*, vol. 48, no. 6, pp. 1410-1422, June 2002.

[9] T.J. Richardson, M.A. Shokrollahi, R.L. Urbanke, "Design of capacity-approaching irregular low-density parity-check codes," *IEEE Trans. on Inf. Theory*, vol. 47, no. 2, pp. 619-637, Feb. 2001.

[10] S. ten Brink, "Designing iterative decoding schemes with the extrinsic information transfer chart," *AEÜ Int. J. Electronic. Commun.*, vol. 54, no. 6, pp. 389-398, Dec. 2000.

[11] S. ten Brink, "Convergence behavior of iteratively decoded parallel concatenated codes," *IEEE Transactions on Communications*, vol. 49, pp. 1727-1737, Oct. 2001.

[12] S.-Y. Chung, T. J. Richardson, and R. Urbanke, "Analysis of sum-product decoding of low-density parity-check codes using a Gaussian Approximation," *IEEE Trans. on Inf. Th*, vol. 47, no. 2, February 2001, pp. 657-670.

[13] H. El Gamal, A.R. Jr. Hammons, "Analyzing the turbo decoder using the Gaussian approximation," *IEEE Trans. on Inf. Theory*, vol. 47, no. 2, pp. 671-686, Feb. 2001.

[14] J. Boutros, G. Caire, "Iterative multiuser joint decoding: unified framework and asymptotic analysis," *IEEE Trans. on Inf. Theory*, vol. 48, no. 7, pp. 1772 -1793, July 2002.

[15] F. Lehmann, G.M. Maggio, "An approximate analytical model of the message passing decoder of LDPC codes", in *Proc. IEEE Int. Symposium Information Theory (ISIT 2002)*, Lausanne, Switzerland, July 2002, p. 31.

[16] M. Ardakani, F.R. Kschischang, "Designing irregular LPDC codes using EXIT charts based on message error rate", in *Proc. IEEE Int. Symposium Information Theory (ISIT 2002)*, Lausanne, Switzerland, July 2002, p. 454.

[17] H. Jin, A. Khandekar, and R. McEliece, "Irregular Repeat-Accumulate Codes," in Proceedings 2nd International Symposium on Turbo codes and Related Topics, Brest, France, Sept. 4, 2000, pp. 1-8 .

[18] J. Boutros, G. Caire, E. Viterbo, H. Sawaya, S. Vialle, "Turbo code at 0.03 dB from capacity limit," in *Proc. IEEE Int. Symposium Information Theory (ISIT 2002)*, Lausanne, Switzerland, July 2002, p. 56.

[19] H. Jin, *Analysis and design of Turbo-like codes*, PhD thesis, California Institute of Technology, May 2001.

[20] A. Ashikhmin, G. Kramer, S. ten Brink, "Extrinsic information transfer functions: A model and two properties", in *Proc. 36th Annual Conference on Information Sciences and Systems (CISS 2002)*, Princeton, New Jersey, March 2002.

[21] A. Ashikhmin, G. Kramer, S. ten Brink, "Code rate and the area under extrinsic information transfer curves", in *Proc. IEEE Int. Symposium Information Theory (ISIT 2002)*, Lausanne, Switzerland, July 2002, p. 115.

[22] S.Y. Chung, *On the construction of some capacity-approaching coding schemes*, PhD thesis, MIT, Sept. 2000.

[23] R. M. Tanner, "A recursive approach to low complexity codes,IEEE Trans. Inform. Theory, vol. IT-27, pp. 533547, Sept. 1981.

[24] J. Pearl, *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*, Morgan Kaufmann, San Mateo, CA, 1988.

[25] R.J. McEliece, D.J.C. MacKay, J-F. Cheng, "Turbo decoding as an instance of Pearl's "belief propagation" algorithm", IEEE Journal on Selected Areas in Communications, vol. 16, pp. 140-152, Feb. 1998.

[26] F.R. Kschischang, B.J. Frey, "Iterative decoding of compound codes by probability propagation in graphical models", IEEE Journal on Selected Areas in Communications, vol. 16, pp. 219-230, Feb. 1998.

[27] D. Forney, "Codes on graphs: normal realizations," *IEEE Trans. on Inf. Theory*, vol. 47, no. 2, pp. 520 -548, Feb. 2001.

[28] L.R. Bahl, J. Cocke, F. Jelinek, J. Raviv, "Optimal decoding of linear codes for minimizing symbol error rate," *IEEE Trans. Inf. Th.*, pp. 284-287, March 1974.

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

[30] S.Y. Chung, G.D. Jr Forney, T.J. Richardson, R. Urbanke, "On the design of low-density parity-check codes within 0.0045 dB of the Shannon limit", IEEE Communications Letters, Vol. 5, pp. 58-60, Feb. 2001.

[31] S. ten Brink, "Exploiting the chain rule of mutual information for the design of iterative decdoing schemes," Proceedings of *the 39th Annual Allerton Conference on Communication, Control, and Computing*, Oct 2001.

[32] M. Tuchler, S. ten Brink, and J. Hagenauer, "Measures for Tracing Convergence of Iterative Decoding Algorithms", in *Proc. 4th International ITG Conference on Source and Channel Coding*, Berlin, Germany, pp. 53-60, Jan 2002.

[33] S. Huettinger, J. Huber, "Extrinsic and intrinsic information in systematic coding", in *Proc. IEEE Int. Symposium Information Theory (ISIT 2002)*, Lausanne, Switzerland, July 2002, p. 116.

[34] T. F. Wong, "Numerical Calculation of Symmetric Capacity of Rayleigh Fading Channel with BPSK/QPSK," *IEEE Communications Letters*, vol. 5, no. 8, pp. 328-330, Aug. 2001.