Finite-Length Analysis of Low-Density Parity-Check Codes on the Binary Erasure Channel

Changyan Di, *Student Member, IEEE*, David Proietti, I. Emre Telatar, *Member, IEEE*, Thomas J. Richardson, and Rüdiger L. Urbanke

Invited Paper

***Abstract*—In this paper, we are concerned with the *finite-length* analysis of low-density parity-check (LDPC) codes when used over the *binary erasure channel* (BEC). The main result is an expression for the *exact average* bit and block erasure probability for a given *regular* ensemble of LDPC codes when decoded *iteratively*. We also give expressions for upper bounds on the average bit and block erasure probability for regular LDPC ensembles and the standard random ensemble under *maximum-likelihood (ML) decoding*. Finally, we present what we consider to be the most important open problems in this area.**

***Index Terms*—Belief propagation, binary erasure channel (BEC), finite-length analysis, low-density parity-check (LDPC) codes.**

I. Introduction

In this paper, we are concerned with the *finite-length* analysis of low-density parity-check (LDPC) codes when used over the *binary erasure channel* (BEC). The main result is an expression for the *exact average* bit and block erasure probability for a given *regular* ensemble $\mathcal{C}(n, x^{\mathtt{l}-1}, x^{\mathtt{r}-1})$ when decoded *iteratively* with message-passing algorithms as in, e.g., [11]. For an introduction into the terminology and basic results of LDPC codes we refer the reader to [3]–[9], [11]–[15].

For a particular code $\mathtt{G}$[1] in a given ensemble $\mathcal{C}(n, \lambda, \rho)$, let $\mathrm{P}_{\mathrm{b}}^{\mathrm{IT}}(\mathtt{G}, \epsilon)$ denote the expected *bit erasure* probability if $\mathtt{G}$ is used to transmit over a BEC with parameter ϵ and if the received word is decoded iteratively by the standard belief propagation decoder. Here, the expectation is over all realizations of the channel. Let $\mathbb{E}_{\mathcal{C}(n, \lambda, \rho)}[\mathrm{P}_{\mathrm{b}}^{\mathrm{IT}}(\mathtt{G}, \epsilon)]$ denote the corresponding *ensemble average*. The following two results are well known, see [7], [9].

[Concentration Around Ensemble Average] For any given $\delta > 0$ there exists an $\alpha(\delta) > 0$ such that

$$\Pr\left\{\left|\mathrm{P}_{\mathrm{b}}^{\mathrm{IT}}(\mathtt{G}, \epsilon) - \mathbb{E}_{\mathcal{C}(n,\lambda,\rho)}\left[\mathrm{P}_{\mathrm{b}}^{\mathrm{IT}}(\mathtt{G}, \epsilon)\right]\right| > \delta\right\} \leq e^{-\alpha(\delta)n}.$$

[Convergence of Ensemble Average to Cycle-Free Case] There exists a constant β such that

$$\left|\mathbb{E}_{\mathcal{C}(n,\lambda,\rho)}\left[\mathrm{P}_{\mathrm{b}}^{\mathrm{IT}}(\mathtt{G}, \epsilon)\right] - \mathbb{E}_{\mathcal{C}(\infty,\lambda,\rho)}\left[\mathrm{P}_{\mathrm{b}}^{\mathrm{IT}}(\mathtt{G}, \epsilon)\right]\right| \leq \frac{\beta}{n}.$$

In words, the first statement asserts that the behavior of the individual codes concentrates around the ensemble average and that this concentration is exponential in the block length. The second statement asserts that the ensemble average converges to the ensemble average of the cycle-free case as the block length tends to infinity.[2] Note, though, that the speed of the convergence to the cycle-free case is known to be of order at least $\frac{1}{n}$ and is likely to be polynomial at best, whereas the converge to the ensemble average is exponential in the block length.[3] The above two statements suggest the following. Fix the block length n and consider individual elements of $\mathcal{C}(n, \lambda, \rho)$. Although the behavior of individual codes can differ significantly from that of the cycle-free (asymptotic) case for moderate block lengths, the behavior of individual instances is likely to be concentrated around the ensemble average. Let us demonstrate this point by means of an example. Consider the situation depicted in Fig. 1. The two solid curves represent $\mathbb{E}_{\mathcal{C}(512, x^2, x^5)}[\mathrm{P}_{\mathrm{b}}^{\mathrm{IT}}(\mathtt{G}, \epsilon)]$ (left solid curve) and $\mathbb{E}_{\mathcal{C}(\infty, x^2, x^5)}[\mathrm{P}_{\mathrm{b}}^{\mathrm{IT}}(\mathtt{G}, \epsilon)]$ (right solid curve), respectively. As we can see, for a block length of $n = 512$, the average bit erasure probability differs significantly from the one of the cycle-free case. Also plotted are curves corresponding to $\mathrm{P}_{\mathrm{b}}^{\mathrm{IT}}(\mathtt{G}, \epsilon)$ for several randomly chosen instances of $\mathcal{C}(512, x^2, x^5)$ (dashed curves). These curves follow the ensemble average very closely for bit erasure probabilities down to 10^{-4}.

From the above observations we can see that the ensemble average plays a significant role in the analysis of finite length codes and that, therefore, computable expressions for

$$\mathbb{E}_{\mathcal{C}(n, \lambda, \rho)}[\mathrm{P}_{\mathrm{b}}^{\mathrm{IT}}(\mathtt{G}, \epsilon)]$$

Manuscript received September 9, 2001; revised January 15, 2002. The material in this paper was presented at the 39th Annual Allerton Conference on Communication, Control and Computing, Allerton, UIUC, October 3–5, 2001.

C. Di, D. Proietti, and R. L. Urbanke are with the Swiss Federal Institute of Technology-Lausanne, LTHC-IC, CH-1015 Lausanne, Switzerland (e-mail: changyan.di@epfl.ch; david.proietti@epfl.ch; rudiger.urbanke@epfl.ch).

I. E. Telatar is with the Swiss Federal Institute of Technology-Lausanne, LTHI-IC, CH-1015 Lausanne, Switzerland (e-mail:emre.telatar @epfl.ch).

T. J. Richardson is with Flarion Technologies, Bedminster, NJ 07921 USA (e-mail: richardson@flarion.com).

Communicated by S. Shamai, Guest Editor.

Publisher Item Identifier S 0018-9448(02)04026-9.

[1] More precisely, $\mathtt{G}$ denotes the bipartite *graph* representing the code.

[2] Recall that in the limit of infinite block length, the support tree up to any fixed given depth of a randomly chosen node or edge is cycle free with probability that goes to one. We, therefore, use the phrases “cycle free” and “infinite block length” interchangeably.

[3] For the erasure channel more precise statements about the convergence speeds can be gained by an analysis of the “error floor,” see [10], [16].

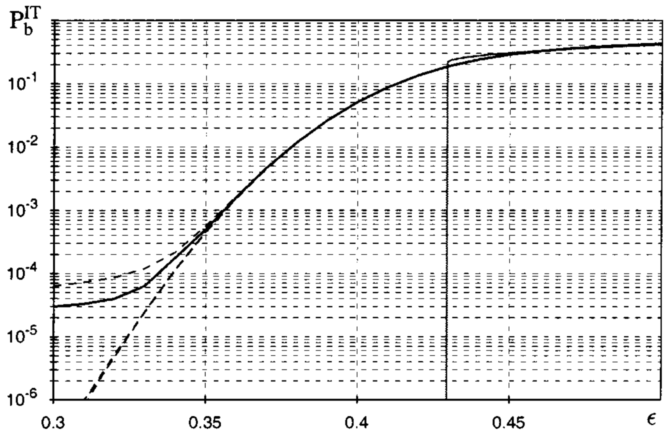

Fig. 1. Concentration of the bit erasure probability $P_b^{IT}(\mathtt{G}, \epsilon)$ for specific instances $\mathtt{G} \in \mathcal{C}(512, x^2, x^5)$ (dashed curves) around the ensemble average $\mathbb{E}_{\mathcal{C}(512, x^2, x^5)}[P_b^{IT}(\mathtt{G}, \epsilon)]$ (left solid curve). (It is noteworthy that there appear to be two dominant modes of behavior.) Also shown is the performance of the cycle-free case, $\mathbb{E}_{\mathcal{C}(\infty, x^2, x^5)}[P_b^{IT}(\mathtt{G}, \epsilon)]$ (right solid curve).

are of considerable value. Viewing the decoding operation from a standard message-passing point of view, it is hard to see how one could derive analytic expressions of $\mathbb{E}_{\mathcal{C}(n, \lambda, \rho)}[P_b^{IT}(\mathtt{G}, \epsilon)]$. Cycles in the graph seem to render the finite-length problem quite intractable. The crucial innovation in this paper is to use as a starting point a *combinatorial* characterization of decoding failures. This combinatorial characterization was originally proposed in [12] in the context of the efficient *encoding* of LDPC codes.

To recall some notation, an ensemble of LDPC codes $\mathcal{C}(n, \lambda, \rho)$ is characterized by its block length n, a variable node degree distribution $\lambda(x) := \sum \lambda_i x^{i-1}$, and a check node degree distribution $\rho(x) = \sum \rho_i x^{i-1}$. Here, $\lambda_i(\rho_i)$ is equal to the probability that a randomly chosen edge is connected to a variable (check) node of degree i. To be specific, consider *regular* ensembles of the form $\mathcal{C}(n, x^{\mathtt{l}-1}, x^{\mathtt{r}-1})$. For example, a typical element of $\mathcal{C}(10, x^2, x^5)$ is shown in Fig. 2. Note that each variable node participates in exactly three checks and that each check node checks exactly six variable nodes.

The following definition characterizes the key object needed to study the finite-length performance of LDPC codes over the BEC.

Definition 1.1 [Stopping Sets]: A *stopping* set $\mathcal{S}$ is a subset of $\mathcal{V}$, the set of variable nodes, such that all neighbors of $\mathcal{S}$ are connected to $\mathcal{S}$ at *least twice.*

As one can see from Fig. 3, for the particular shown $\mathtt{G}$ the set $\{\mathtt{v}_1, \mathtt{v}_2, \mathtt{v}_3, \mathtt{v}_4\}$ is a stopping set.

Note, in particular, that the empty set is a stopping set. The space of stopping sets is closed under unions, i.e., if $\mathcal{S}_1$ and $\mathcal{S}_2$ are both stopping sets then so is $\mathcal{S}_1 \cup \mathcal{S}_2$. (To see this note that if $\mathtt{c}$ is a neighbor of $\mathcal{S}_1 \cup \mathcal{S}_2$ then it must be a neighbor of at least one of $\mathcal{S}_1$ or $\mathcal{S}_2$, assume that $\mathtt{c}$ is a neighbor of $\mathcal{S}_1$. Since $\mathcal{S}_1$ is a stopping set, $\mathtt{c}$ has at least two connections to $\mathcal{S}_1$ and therefore at least two connections to $\mathcal{S}_1 \cup \mathcal{S}_2$.) Each subset of $\mathcal{V}$ thus clearly contains a unique maximal stopping set (which might be the empty set).

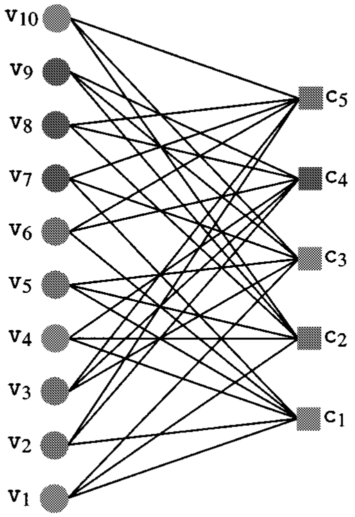

Fig. 2. A specific element $\mathtt{G}$ from the ensemble $\mathcal{C}(10, x^2, x^5)$.

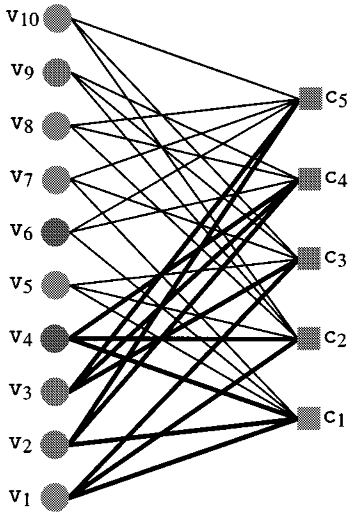

Fig. 3. The set $\{\mathtt{v}_1, \mathtt{v}_2, \mathtt{v}_3, \mathtt{v}_4\}$ is a stopping set.

The next lemma shows the crucial role that stopping sets play in the process of iterative decoding of LDPC codes when used over the BEC.

Lemma 1.1 [Combinatorial Characterization of Iterative Decoder Performance]: Let $\mathtt{G}$ be a given element from $\mathcal{C}(n, \lambda, \rho)$. Assume that we use $\mathtt{G}$ to transmit over the BEC and that we decode the received word in an iterative fashion until either the codeword has been recovered or until the decoder fails to progress further. Let $\mathcal{E}$ denote the subset of the set of variable nodes which is erased by the channel. Then the set of erasures which remain when the decoder stops is equal to the unique maximal stopping set of $\mathcal{E}$.

Proof: Let $\mathcal{S}$ be a stopping set contained in $\mathcal{E}$. We claim that the iterative decoder cannot determine the variable nodes contained in $\mathcal{S}$. This is true, since even if all other bits were known, every neighbor of $\mathcal{S}$ has at least two connections to the set $\mathcal{S}$ and so all messages to $\mathcal{S}$ will be erasure messages. It follows that the decoder cannot determine the variables contained in the unique maximal stopping set contained in $\mathcal{E}$. Conversely, if the decoder terminates at a set $\mathcal{S}$, then all messages entering this subset must be erasure messages which happens only if all neighbors of $\mathcal{S}$ have at least two connections to $\mathcal{S}$. In other words, $\mathcal{S}$ must be a stopping set and, since no erasure contained in a stopping set can be determined by the iterative decoder, it must be the maximal such stopping set. □

In order now to determine the exact (block) erasure probability under iterative decoding it remains to find the probability that a random subset of the set of variable nodes (the set of "erasures") of a randomly chosen element from the ensemble $\mathcal{C}(n, x^{\mathtt{l}-1}, x^{\mathtt{r}-1})$ contains a nonempty stopping set. We show in Theorem 2.1 that this can be done *exactly*. In Section III, we consider the maximum likelihood (ML) performance of LDPC ensembles as well as of the standard random ensemble. It is instructive to study the ML performance since this makes it possible to distinguish how much of the incurred performance loss of iterative coding systems is due to the suboptimal decoding and how much is due to the particular choice of codes. Finally, in Section IV, we present what we consider to be the most important open problems in this area.

II. Finite-Length Analysis

A. LDPC Codes Under Belief Propagation Decoding

The characterization of decoding failures stated in Lemma 1.1 reduces the task of the exact determination of the performance of iterative decoders to a combinatorial problem. In this section, we present a solution to that combinatorial problem. In the sequel, if $f(x)$ is a power series, $f(x) = \sum_{i\geq 0} f_i x^i$, we denote by $\operatorname{coef}(f(x), x^i)$ its ith coefficient f_i.

Theorem 2.1: Let $\mathrm{P}_{\mathrm{b}}^{\mathrm{IT}}(\mathtt{G}, \epsilon)$ denote the *bit erasure* probability when transmitting over a BEC with erasure probability ϵ using a code $\mathtt{G}$, $\mathtt{G} \in \mathcal{C}(n, x^{\mathtt{l}-1}, x^{\mathtt{r}-1})$, and a belief propagation decoder. Hereby we assume that we iterate until either all erasures have been determined or the decoder fails to progress further. In a similar manner, let $\mathrm{P}_{\mathrm{B}}^{\mathrm{IT}}(\mathtt{G}, \epsilon)$ denote the *block erasure* probability. Define the functions $T(v, c, d)$, $N(v, c, d)$, $M(v, c, d)$, and $O(v, s, c, d)$ by the recursions

$$T(v, c, d) := \binom{d + c\mathtt{r}}{v\mathtt{l}} (v\mathtt{l})! \tag{2.1}$$

$$N(v, c, d) := T(v, c, d) - M(v, c, d) \tag{2.2}$$

$$M(v, c, d) := \sum_{s>0} \binom{v}{s} O(v, s, c, d) \tag{2.3}$$

$$\begin{aligned} O(v, s, c, d) := \sum_{k} \binom{c}{k} &\operatorname{coef}(((1+x)^{\mathtt{r}} - 1 - \mathtt{r}x)^k \\ &\cdot (1+x)^d, x^{s\mathtt{l}})(s\mathtt{l})! \\ &N(v - s, c - k, d + k\mathtt{r} - s\mathtt{l}) \end{aligned} \tag{2.4}$$

and the boundary condition

$$O(v, s, c, d) = 0, \qquad \text{if } s \leq 0 \text{ or } v\mathtt{l} > c\mathtt{r} + d.$$

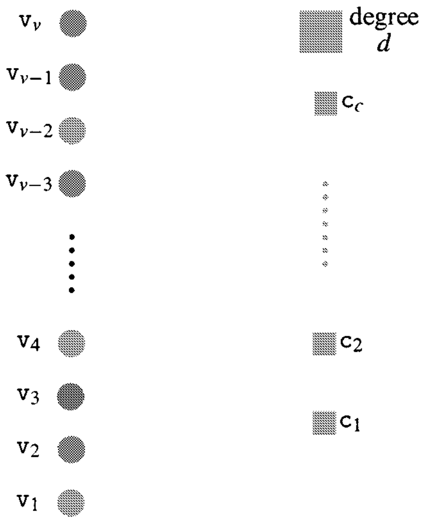

Fig. 4. There are v variable nodes of degree $\mathtt{l}$, c check nodes of degree $\mathtt{r}$, and one *super* check node of degree d.

(Note, in (2.4), that $\operatorname{coef}(((1+x)^{\mathtt{r}} - 1 - \mathtt{r}x)^k (1+x)^d, x^{s\mathtt{l}}) = 0$ if $d + k\mathtt{r} - s\mathtt{l} < 0$ so $N(v - s, c - k, d + k\mathtt{r} - s\mathtt{l})$ need not be defined for this case.) Then

$$\begin{aligned} &\mathbb{E}_{\mathcal{C}(n, x^{\mathtt{l}-1}, x^{\mathtt{r}-1})}\left[\mathrm{P}_{\mathrm{b}}^{\mathrm{IT}}(\mathtt{G}, \epsilon)\right] \\ &\quad = \sum_{e} \binom{n}{e} \epsilon^e (\bar{\epsilon})^{n-e} \sum_{s} \frac{s}{n} \frac{\binom{e}{s} O\left(e, s, n\frac{\mathtt{l}}{\mathtt{r}}, 0\right)}{T\left(e, n\frac{\mathtt{l}}{\mathtt{r}}, 0\right)}, \\ &\mathbb{E}_{\mathcal{C}(n, x^{\mathtt{l}-1}, x^{\mathtt{r}-1})}\left[\mathrm{P}_{\mathrm{B}}^{\mathrm{IT}}(\mathtt{G}, \epsilon)\right] \\ &\quad = \sum_{e} \binom{n}{e} \epsilon^e (\bar{\epsilon})^{n-e} \sum_{s} \frac{\binom{e}{s} O\left(e, s, n\frac{\mathtt{l}}{\mathtt{r}}, 0\right)}{T\left(e, n\frac{\mathtt{l}}{\mathtt{r}}, 0\right)} \\ &\quad = \sum_{e=0}^{n\frac{\mathtt{l}}{\mathtt{r}}-1} \binom{n}{e} \epsilon^e (\bar{\epsilon})^{n-e} \left(1 - \frac{N\left(e, n\frac{\mathtt{l}}{\mathtt{r}}, 0\right)}{T\left(e, n\frac{\mathtt{l}}{\mathtt{r}}, 0\right)}\right) \\ &\qquad + \sum_{e=n\frac{\mathtt{l}}{\mathtt{r}}}^{n} \binom{n}{e} \epsilon^e (\bar{\epsilon})^{n-e} \end{aligned}$$

where $\bar{\epsilon} := 1 - \epsilon$.

Proof: Consider the situation depicted in Fig. 4. There are v variable nodes of degree $\mathtt{l}$, c check nodes of degree $\mathtt{r}$, and one *super* check node of degree d.[4] Label the $v\mathtt{l}$ variable node sockets in some arbitrary but fixed way with elements from the set $[v\mathtt{l}] := \{1, 2, \ldots, v\mathtt{l}\}$ and, in a similar manner, label the $c\mathtt{r} + d$ check node sockets in some arbitrary but fixed way with elements from the set $[c\mathtt{r} + d]$. Let

$$\alpha\colon [v\mathtt{l}] \to [v], \quad \beta\colon [c\mathtt{r} + d] \to [c + 1]$$

denote maps which describe the association of variable and check node sockets to their respective nodes, so that, e.g., if $\alpha(3) = 5$ then this signifies that the third variable node socket emanates from the fifth variable node. We always label the c *regular* check nodes by $[c]$ and set the label of the super check node to $(c + 1)$.

[4] As we will see shortly, it is the introduction of this extra check node which makes it possible to write down the recursions.

For simplicity, we will refer to a particular realization of connecting the $v\mathtt{l}$ variable node sockets to the $c\mathtt{r}+d$ check node sockets as a *constellation*. More precisely, a constellation is an injective map (so $v\mathtt{l} \leq c\mathtt{r}+d$ is required)

$$\tau\colon [v\mathtt{l}] \rightarrow [c\mathtt{r}+d]$$

so that variable node socket i, $i \in [v\mathtt{l}]$, is connected to check node socket $\tau(i)$, $\tau(i) \in [c\mathtt{r}+d]$. Let $\mathcal{T}(v, c, d)$ denote the set of all such maps and let $T(v, c, d) := |\mathcal{T}(v, c, d)|$. Since there are $\binom{c\mathtt{r}+d}{v\mathtt{l}}$ degrees of freedom in choosing which of the check node sockets are connected and a further $(v\mathtt{l})!$ ways of permuting the corresponding edges, $T(v, c, d)$ is as given in (2.1).

We will say that a constellation contains a *stopping set* $\mathcal{S}$ if it contains a nonempty subset of the variable nodes such that any *regular* check node $\mathtt{c}$ which is connected to $\mathcal{S}$, is connected to $\mathcal{S}$ *at least twice*. More precisely, $\mathcal{S}$, $\mathcal{S} \subseteq \mathcal{V}$, is a stopping set if

$$|\{i \in [v\mathtt{l}]\colon \alpha(i) \in \mathcal{S}; \beta(\tau(i)) = j\}| \neq 1, \qquad \forall j \in [c].$$

Note that this definition is slightly more general than the one given in Definition 1.1 since in our current setup we have in addition a super check node of degree d. In particular, in this extended definition, *no restrictions* are placed on the number of connections from the stopping set $\mathcal{S}$ to the super check node.

Clearly, the set $\mathcal{T}(v, c, d)$ can be partitioned into the set of maps that contain *no* stopping set, call this set $\mathcal{N}(v, c, d)$, and the set of maps that contain *at least one* stopping set, call this set $\mathcal{M}(v, c, d)$. Letting

$$N(v, c, d) = |\mathcal{N}(v, c, d)|$$

and

$$M(v, c, d) = |\mathcal{M}(v, c, d)|$$

we, therefore, have the relationship (2.2).

Consider now $\mathcal{M}(v, c, d)$, the set of constellations that contain at least one stopping set. Observe that if $\mathcal{S}_1$ and $\mathcal{S}_2$ are two stopping sets then their *union* is a stopping set. It follows that each element of $\mathcal{M}(v, c, d)$ contains a *unique maximal* stopping set. Therefore, we have

$$\mathcal{M}(v, c, d) = \bigcup_{\mathcal{S} \subseteq [v]} \mathcal{O}(v, \mathcal{S}, c, d) \tag{2.5}$$

where $\mathcal{O}(v, \mathcal{S}, c, d)$ denotes the set of constellations which have $\mathcal{S}$ as their unique maximal stopping set. By some abuse of notation, let

$$O(v, |\mathcal{S}|, c, d) = |\mathcal{O}(v, \mathcal{S}, c, d)|$$

where we have used the fact that the cardinality of $\mathcal{O}(v, \mathcal{S}, c, d)$ only depends on the cardinality of $\mathcal{S}$ but not on the specific choice of variable nodes. Since there are $\binom{v}{s}$ choices of $\mathcal{S}$ of size s and since the union in (2.5) is disjoint we get (2.3).

It remains to prove the recursion (2.4) which links $O(v, s, c, d)$ to $N(v, c, d)$. Consider the situation depicted in Fig. 5, where a specific set $\mathcal{S}$ of cardinality s is chosen. We are interested in counting the elements of $\mathcal{O}(v, \mathcal{S}, c, d)$. Note that by definition of $\mathcal{O}(v, \mathcal{S}, c, d)$, $\mathcal{S}$ is the unique maximal stopping set. First, this implies that $\mathcal{S}$ is a stopping set. Consider those elements of $\mathcal{O}(v, \mathcal{S}, c, d)$ for which the set $\mathcal{S}$ is connected to k (out of the c) regular check nodes. There are $\binom{c}{k}$ ways of choosing these check nodes. Next, there are

$$\text{coef}(((1+x)^{\mathtt{r}} - 1 - \mathtt{r}x)^k (1+x)^d, x^{s\mathtt{l}})$$

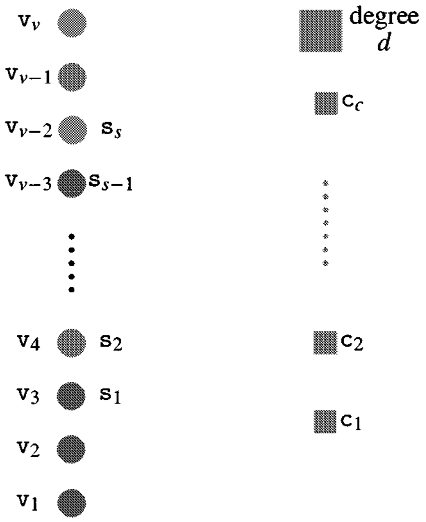

Fig. 5. There are v variable nodes of degree $\mathtt{l}$, c check nodes of degree $\mathtt{r}$, and one *super* check node of degree d. Further, $\mathcal{S}$ is a subset of $\mathcal{V}$, the set of variable nodes, of cardinality s.

ways of choosing the check node sockets to which the $s\mathtt{l}$ sockets of the set $\mathcal{S}$ are connected. Finally, the $s\mathtt{l}$ edges emanating from $\mathcal{S}$ can be permuted in $(s\mathtt{l})!$ ways.

So far we have only been concerned with edges that emanate from $\mathcal{S}$. We still need to ensure that we only count those constellations for which $\mathcal{S}$ is the *maximal* stopping set. Consider a set $\mathcal{L} \subseteq \mathcal{V}\backslash\mathcal{S}$. Assume that $\mathcal{L}$ has the property that any regular check node which is connected to $\mathcal{L}$ but *not* to $\mathcal{S}$ is connected to $\mathcal{L}$ at least twice. Then clearly $\mathcal{S} \cup \mathcal{L}$ is also a stopping set and so $\mathcal{S}$ is not the maximal stopping set. Conversely, assume that $\mathcal{S}$ is not the maximal stopping set. Let $\mathcal{K}$ be the maximal stopping set and consider $\mathcal{L} := \mathcal{K}\backslash\mathcal{S}$. By definition, every regular check node which is connected to $\mathcal{K}$ is connected to $\mathcal{K}$ at least twice. Therefore, every regular check node which is connected to $\mathcal{L}$ but not to $\mathcal{S}$ is connected to $\mathcal{L}$ at least twice. We conclude that $\mathcal{S}$ will be the unique maximal stopping set iff $\mathcal{V}\backslash\mathcal{S}$ does not contain a subset $\mathcal{L}$ with the property that every regular check node which is connected to $\mathcal{L}$ but not to $\mathcal{S}$ is connect to $\mathcal{L}$ at least twice. How many constellations are there which fulfill this property? A moment's thought shows that this number is equal to $N(v-s, c-k, d+k\mathtt{r}-s\mathtt{l})$: there are $v-s$ variable nodes in $\mathcal{V}\backslash\mathcal{S}$; there are further $c-k$ regular check nodes which are not neighbors of $\mathcal{S}$; and the remaining $d+k\mathtt{r}-s\mathtt{l}$ available sockets can be combined relegated the super check node.

The bit erasure and block erasure probability can be expressed in a straightforward manner in terms of $O(v, s, c, 0)$. The decoder terminates in the unique maximal stopping set contained in the set of erased bits. If we are interested in the average fraction of erased bits remaining, then a maximal stopping set of size s will cause s erasures. If we are interested in the block erasure probability then each nonempty stopping set counts equally. From these observations the stated formulas for the erasure probabilities follow in a straightforward manner. For the second expression giving the block erasure probability we argue as follows: the quantity

$$\left(1 - \frac{N(e, n\frac{\mathtt{l}}{\mathtt{r}}, 0)}{T(e, n\frac{\mathtt{l}}{\mathtt{r}}, 0)}\right)$$

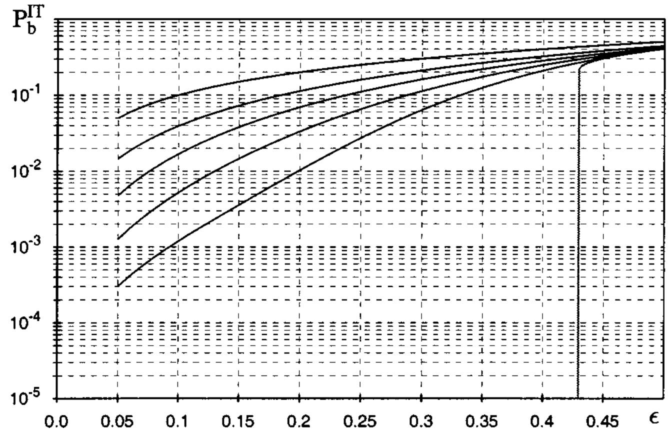

Fig. 6. $\mathbb{E}_{\mathcal{C}(n,\, x^2,\, x^5)}[\mathrm{P}_b^{\mathrm{IT}}(\mathtt{G},\, \epsilon)]$ as a function of ϵ for $n = 2^i$, $i \in [5]$. Also shown is the limit $\mathbb{E}_{\mathcal{C}(\infty,\, x^2,\, x^5)}[\mathrm{P}_b^{\mathrm{IT}}(\mathtt{G},\, \epsilon)]$ (cycle-free case).

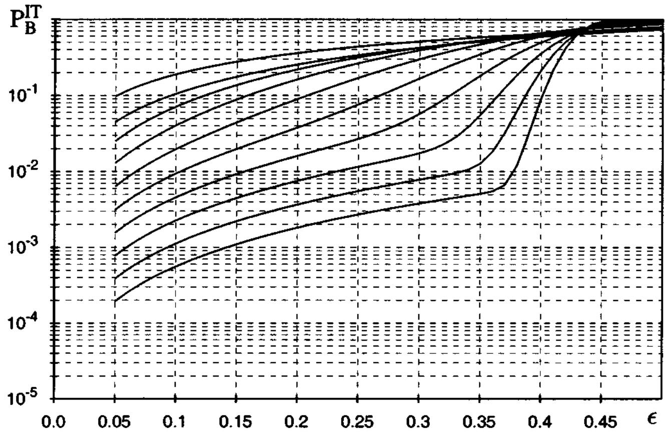

Fig. 7. $\mathbb{E}_{\mathcal{C}(n,\, x^2,\, x^5)}[\mathrm{P}_B^{\mathrm{IT}}(\mathtt{G},\, \epsilon)]$ as a function of ϵ for $n = 2^i$, $i \in [10]$.

is the probability that a randomly chosen subset of size e contains a nonempty stopping set. If we multiply this quantity with the probability that the size of the erasure set is equal to e and sum over all e then we get the block erasure probability. We can simplify the expression by verifying that this quantity is equal to one if $e \geq n\frac{1}{\mathtt{r}}$. □

Example 1: Consider the ensemble $\mathcal{C}(n,\, x^2,\, x^5)$. Fig. 6 shows $\mathbb{E}_{\mathcal{C}(n, x^2, x^5)}[\mathrm{P}_b^{\mathrm{IT}}(\mathtt{G},\, \epsilon)]$ as a function of ϵ for $n = 2^i$, $i \in [5]$. Also shown is the limit $\mathbb{E}_{\mathcal{C}(\infty, x^2, x^5)}[\mathrm{P}_b^{\mathrm{IT}}(\mathtt{G}, \epsilon)]$ (cycle-free case). This limiting curve can be determined in the following way. Recall that the *threshold* ϵ^* associated to a degree distribution pair $(\lambda,\, \rho)$ can be characterized as[5]

$$\epsilon^* := \sup\{\epsilon \geq 0\colon \epsilon\lambda(1 - \rho(1 - x)) - x < 0,\ \forall x \in (0,\, 1]\}.$$

Assume now that the initial erasure probability ϵ is strictly above this threshold ϵ^*. In this case, the decoder will not terminate successfully and a fixed fraction of erasures will remain. To determine this fraction define $x(\epsilon)$, where $\epsilon > \epsilon^*$, as

$$x(\epsilon) := \sup\{0 \leq x \leq \epsilon\colon \epsilon\lambda(1 - \rho(1 - x)) - x = 0\}.$$

In words, $x(\epsilon)$ is the erasure probability of the messages emitted from the variable nodes at the point where the decoder terminates. To this corresponds an erasure probability of the messages emitted by the check nodes of $1 - \rho(1 - x(\epsilon))$. From this quantity it is now easy to see that the corresponding bit erasure probability is equal to $\epsilon L(1 - \rho(1 - x(\epsilon)))$, where

$$L(x) := \frac{\int_0^x \lambda(z)\, dz}{\int_0^1 \lambda(z)\, dz}$$

is the variable node degree distribution from the node perspective. Therefore, the limiting curve is given in parametric form as

$$(\epsilon,\, \epsilon L(1 - \rho(1 - x(\epsilon)))), \qquad \epsilon \geq \epsilon^*.$$

For the specific example of the $(3,\, 6)$-regular code it is more convenient to parameterize the curve by x (instead of ϵ). We know from [1] that $\epsilon^* = 0.42944$ and the corresponding $x(\epsilon^*)$ is given by $x(\epsilon^*) = 0.260399$. From $\epsilon(1 - (1 - x)^5)^2 = x$ we get

$$\epsilon(x) = \frac{x}{(1 - (1 - x)^5)^2}, \qquad x \geq x(\epsilon^*)$$

so that the limit curve is given in parametric form by

$$\left(\frac{x}{(1 - (1 - x)^5)^2},\, x(1 - (1 - x)^5)\right), \qquad x \geq x(\epsilon^*).$$

B. *Efficient Evaluation of Expressions*

It is clear that the recursions stated in Theorem 2.1 quickly become impractical to evaluate as the block length grows (this is in fact the reason why in Fig. 6 we only depicted the curves up to length 32!). For the cases $\mathtt{l} = 2$ or $\mathtt{l} = 3$ the following recursions are substantially easier to evaluate.

Fig. 7 shows the average block erasure probability for the ensemble $\mathcal{C}(n,\, x^2,\, x^5)$ for block lengths $n = 2^i$, $i \in$, as determined by the following expressions.

Theorem 2.2: Let $a_2(v,\, u,\, s,\, d)$ and $a_3(v,\, u,\, s,\, d)$ be recursively defined by

$$\begin{aligned} a_2(v,\, u,\, s,\, d)s = {} & a_2(v-1,\, u-1,\, s-1,\, d) \\ & \cdot (u\mathtt{r} + (1-\mathtt{r})s - v\mathtt{l} + \mathtt{l} - 1 + d) \\ & + a_2(v-1,\, u-1,\, s,\, d)s \\ & + a_2(v-1,\, u-2,\, s-2,\, d), \\ & \qquad v \geq 1,\, u \geq 0,\, s \geq 1 \end{aligned} \tag{2.6}$$

and

$$\begin{aligned} a_3(v,\, u,\, s,\, d)s = {} & a_3(v-1,\, u-3,\, s-3,\, d)\frac{1}{2} \\ & + a_3(v-1,\, u-2,\, s-2,\, d) \\ & \cdot ((u-s)\mathtt{r} - (v-1)\mathtt{l} + s - 2 + d) \\ & + a_3(v-1,\, u-2,\, s-1,\, d)((s-1) + 1/2) \\ & + a_3(v-1,\, u-1,\, s-1,\, d) \\ & \cdot \binom{(u-s)\mathtt{r} - (v-1)\mathtt{l} + s - 1 + d}{2} \end{aligned}$$

[5] Note, that the range of x in this definition can be chosen to be $x \in (0,\, 1]$ rather than $x \in (0,\, \epsilon]$ since for $x \in (\epsilon,\, 1]$ the inequality is automatically fulfilled if it is fulfilled for $x = \epsilon$.

$$\begin{aligned}
&+a_3(v-1,\, u-1,\, s,\, d)\frac{s}{\mathtt{r}-1}\\
&\cdot\binom{(u-s)\mathtt{r}-(v-1)\mathtt{l}+s-1+d}{2}\\
&-a_3(v-1,\, u-1,\, s+d)\frac{s}{\mathtt{r}-1}\\
&+\binom{(u-s-1)\mathtt{r}-(v-1)\mathtt{l}+s+d}{2}\\
&+a_3(v-1,\, u-1,\, s+1+d)\binom{s+1}{2}
\end{aligned}$$

with the boundary condition

$$a(v=0,\, u,\, s,\, d)=\begin{cases}1, & u=0,\, s=0,\, d\geq 0\\ 0, & \text{otherwise.}\end{cases}$$

Define

$$a(v,\, u,\, d):=\sum_s \frac{a(v,u,s,d)}{(\mathtt{r}-1)^s}.$$

Then

$$N(v,\, c,\, d):=(\mathtt{l}!)^v v!\sum_u a(v,\, u,\, d)\frac{c!}{(c-u)!}\,\mathtt{r}^u(\mathtt{r}-1)^u.$$

The basic idea in deriving these recursions is simple although the details become quite cumbersome. Consider a constellation which does not contain a stopping set. Then it must contain at least one degree-one check node. Peal off this check node, i.e., remove it together with its connected variable node, any edges connected to these nodes and any further check nodes which, after removal of these edges, have degree zero. The result will be a smaller constellation which again does not contain a stopping set and so we can apply this procedure recursively. Reversing this procedure, we see that constellations which do not contain stopping sets can be built up one variable node at a time. This gives rise to the stated recursions. Some care has to be taken to make sure that we count each constellation only once since in general constellations might contain more than just one check node of degree one and so the same constellation can be constructed in general in many ways starting from suitable smaller constellations. In the above recursions, v denotes the number of variable nodes of a constellation, u the number of used check nodes, s the number of check nodes of degree one, and d the degree of the super check node.

In more detail: Consider a stopping-set-free constellation which has v variable nodes, uses u check nodes, s of which have degree one, and is labeled by the *standard* labels $[v]$ and $[u]$, respectively. Let $\mathcal{A}(v,\, u,\, s)$ denote the set of all such constellations and let $\alpha(v,\, u,\, s)$ denote its cardinality. We will now describe how we can *prune* and *grow* constellations. This will give rise to the desired recursion. Fix an element from $\mathcal{A}(v,\, u,\, s)$. For each variable node, call it $\mathtt{v}$, $\mathtt{v}\in[v]$, let $m=m(\mathtt{v})$ denote the number of neighboring check nodes of degree one. We will call m the *multiplicity* of $\mathtt{v}$ and we will denote these neighbors by $\mathtt{c}_1,\, \ldots,\, \mathtt{c}_m$. To *prune* an element of $\mathcal{A}(v,\, u,\, s)$, pick a variable node $\mathtt{v}$ of multiplicity at least one and delete $\mathtt{v}$ and $\mathtt{c}_1,\, \ldots,\, \mathtt{c}_m$ from the constellation. The parameters of the new constellation are therefore $v'=v-1$, $u'=u-m$, and s'. In order to make this constellation an element of $\mathcal{A}(v-1,\, u-1,\, s')$ we have to ensure that its labeling is the standard one with label sets $[v-1]$ and $[u-m]$ for the variable and check nodes, respectively. We do this in the natural way, i.e., for the pruned constellation all labels smaller than $\mathtt{v}$ remain unchanged whereas all labels larger than $\mathtt{v}$ are decreased by one. An equivalent procedure is applied at the check node side where we have deleted m nodes.

The above procedure can be inverted, i.e., if we start with this pruned constellation and add a variable node with label $\mathtt{v}$ as well as m check nodes with labels $\mathtt{c}_1,\, \ldots,\, \mathtt{c}_m$ then we can recover our original constellation by connecting the edges in an appropriate way. Hereby, in adding, e.g., the variable node with label $\mathtt{v}$ we have to increase all labels of variable nodes with labels equal to at least $\mathtt{v}$ by exactly one and a similar remark applies for the check nodes. Let $\mathcal{A}^{\mathtt{v},\,\mathtt{c}_1,\,\ldots,\,\mathtt{c}_m}(v,\, u,\, s)$ denote the subset of $\mathcal{A}(v,\, u,\, s)$ which contain the variable node $\mathtt{v}$ of multiplicity m which is connected to the degree-one check nodes $\mathtt{c}_1,\, \ldots,\, \mathtt{c}_m$. Now note that each element in $\mathcal{A}^{\mathtt{v},\,\mathtt{c}_1,\,\ldots,\,\mathtt{c}_m}(v,\, u,\, s)$ can be reconstructed in a unique way from an element of $\bigcup_{s'}\mathcal{A}(v-1,$ $u-m,\, s')$ by adding $\mathtt{v}$ and $\mathtt{c}_1,\, \ldots,\, \mathtt{c}_m$. It follows that a given element of $\mathcal{A}(v,\, u,\, s)$ can be reconstructed in exactly as many ways as the number of its variable nodes which have multiplicity at least one. Note that, by definition, the sum of the multiplicities of all variable nodes is equal to s. Therefore, the above statement can be rephrased in the following manner. If we weigh each reconstruction by the multiplicity of the inserted variable node then this weighted sum of reconstructions equals s.

Consider now the recursion for $\mathtt{l}=2$ in more detail. Without much loss of generality we assume here that $d=0$, i.e., that there is no super check node. The general case is a quite straightforward extension. On the left-hand side of the recursion we write $\alpha_2(v,\, u,\, s)s$, which by our remarks above is equal to the *weighted* sum of reconstructions. There are only three possible ways of reaching an element of $\mathcal{A}(v,\, u,\, s)$ by adding one variable node of degree two to a constellation in $\bigcup_{u',\,s'}\mathcal{A}(v-1,$ $u',\, s')$. We can have

$$\begin{gathered}(u'=u-1,\, s'=s-1)\\ (u'=u-1,\, s'=s)\end{gathered}$$

or

$$(u'=u-2,\, s'=s-2).$$

Consider first the case $(u'=u-1,\, s'=s-1)$, and therefore $m=1$. In this case, we can choose the label $\mathtt{v}$ in v ways and the label $\mathtt{c}_1$ in u ways. Further, there are $\mathtt{r}$ choices for the socket of $\mathtt{c}_1$ and, as a moment's thought shows, $u\mathtt{r}+(1-\mathtt{r})s-v\mathtt{l}+\mathtt{l}-1$ choices for the socket of the second edge. Next, look at the case $(u'=u-1,\, s'=s)$ which also implies that $m=1$. As before, we can choose the label $\mathtt{v}$ in v ways, the label $\mathtt{c}_1$ in u ways, and there are $\mathtt{r}$ choices for the socket of $\mathtt{c}_1$. The second edge is now connected to a check node of degree one, and there are $s'=s$ of them and further we can choose one out of $\mathtt{r}-1$ sockets. Finally, consider the case $(u'=u-2,\, s'=s-2)$, which implies that $m=2$. As before, we can choose the label $\mathtt{v}$ in v ways and the labels $\mathtt{c}_1$, $\mathtt{c}_2$ in $\binom{u}{2}$ ways and we have $\mathtt{r}^2$ choices for the sockets of the two check nodes. Since we count weighted reconstructions we also have to add a factor 2. In summary we get the recursion

$$\begin{aligned}\alpha_2(v, u, s)s = {}&\alpha_2(v-1, u-1, s-1)\\ &\cdot vu\mathtt{r}(u\mathtt{r}+(1-\mathtt{r})s-v\mathtt{l}+\mathtt{l}-1)\\ &+\alpha_2(v-1, u-1, s)vu\mathtt{r}s(\mathtt{r}-1)\\ &+\alpha_2(v-1, u-2, s-2)v\binom{u}{2}\mathtt{r}^2 2,\\ &\qquad u \geq 0,\ s \geq 1.\end{aligned}$$

We can simplify the above recursions by noting that several factors are common to all terms and only depend on v and u. This gives rise to the recursion stated in (2.6).

Rather than explaining the case $\mathtt{l}=3$ in detail we refer the reader to [16], where the above approach has been generalized to arbitrary $\mathtt{l}$ and a systematic derivation is given.

There are many more alternative ways in which expressions for the average block or bit erasure probability can be derived. We mention one which is particular to the case $\mathtt{l}=2$. Note that in this case, a stopping-set-free constellation cannot contain a double edge, i.e., each variable node connects two distinct check nodes. Therefore, stopping-set-free constellations can be represented as regular graphs, whose nodes are the check nodes of the bipartite graph and whose edges are in one-to-one correspondence with variable nodes of the bipartite graph. A moment's thought now shows that stopping-set-free constellations on the bipartite graph correspond to a *forest* in the corresponding regular graph. We can, therefore, equivalently count the number of forests, where each node in the regular graph has degree at most $\mathtt{r}$ and where sockets and edges are labeled.

III. ML Decoding

It is instructive to compare the performance of an LDPC ensemble under *iterative decoding* to that of the same LDPC ensemble under *ML decoding* as well as the performance of the standard random ensemble under *ML decoding*. The reason for our interest in these quantities is that they indicate how much of the performance loss of iterative coding systems is due to the choice of codes and how much is due to the choice of the suboptimal decoding algorithm. We note that we assume an ML decoder which determines all those bits which are uniquely specified by the channel observations but does not break ties and therefore we will deal with true erasure probabilities rather than error probabilities.

A. Standard Random Ensemble Under ML Decoding

Theorem 3.1: Consider the ensemble $\mathcal{C}(n, k)$ of binary linear codes of length n and dimension k defined by means of their parity-check matrix H, where each entry of H is an independent and identically distributed (i.i.d.) Bernoulli random variable with parameter one-half. Let $\mathrm{P}_{\mathrm{b}}^{\mathrm{ML}}(H, \epsilon)$ denote the *bit erasure probability* of a particular code defined by the parity-check matrix H when used to transmit over a BEC with erasure probability ϵ and when decoded by an ML decoder. Let $\mathrm{P}_{\mathrm{B}}^{\mathrm{ML}}(H, \epsilon)$ denote the corresponding *block erasure probability*. Then

$$\begin{aligned}&\mathbb{E}_{\mathcal{C}(n,k)}[\mathrm{P}_{\mathrm{b}}^{\mathrm{ML}}(H, \epsilon)]\\ &\quad=\sum_{e=0}^{n}\binom{n}{e}\epsilon^e\bar{\epsilon}^{n-e}\frac{e}{n}\frac{\sum_j R(e-1, n-k, j)2^j}{2^{(n-k)e}} \qquad (3.1)\end{aligned}$$

$$\begin{aligned}&\mathbb{E}_{\mathcal{C}(n,k)}[\mathrm{P}_{\mathrm{B}}^{\mathrm{ML}}(H, \epsilon)]\\ &\quad=\sum_{e=0}^{n-k}\binom{n}{e}\epsilon^e\bar{\epsilon}^{n-e}\left[1-\prod_{i=0}^{e-1}(1-2^{i-n+k})\right]\\ &\qquad+\sum_{e=n-k+1}^{n}\binom{n}{e}\epsilon^e\bar{\epsilon}^{n-e} \qquad (3.2)\end{aligned}$$

where $R(l, m, k)$ is the number of $l \times m$ binary matrices of rank k. An enumeration is given in Appendix A.

Proof: First consider the block erasure probability. Let $\mathcal{E}$ denote the set of erasures and let $H_{\mathcal{E}}$ denote the submatrix of H which consists of those columns of H which are indexed by $\mathcal{E}$. In a similar manner, let $x_{\mathcal{E}}$ denote those components of the codeword x which are indexed by $\mathcal{E}$. From the defining equation $Hx^T = 0^T$ we conclude that

$$H_{\mathcal{E}}x_{\mathcal{E}}^T = H_{\overline{\mathcal{E}}}x_{\overline{\mathcal{E}}}^T \qquad (3.3)$$

where $\overline{\mathcal{E}} := [n]\backslash\mathcal{E}$. Now note that if x denotes the transmitted codeword and $\mathcal{E}$ denotes the set of erasures then $s^T := H_{\overline{\mathcal{E}}}x_{\overline{\mathcal{E}}}^T$, the right-hand side of (3.3), is *known* to the receiver. In standard terminology, s is called the *syndrome*. Consider now the equation $H_{\mathcal{E}}x_{\mathcal{E}}^T = s^T$. Since, by assumption, x is a valid codeword, we know that this equation has *at least one* solution. It has *multiple* solutions, i.e., the ML decoder will not be able to recover the codeword uniquely, iff $H_{\mathcal{E}}$ has rank less than $|\mathcal{E}|$. From (A1) we know that this happens with probability

$$1-\frac{R(|\mathcal{E}|, n-k, |\mathcal{E}|)}{2^{(n-k)|\mathcal{E}|}} = \begin{cases}1-\prod_{i=0}^{|\mathcal{E}|-1}(1-2^{i-n+k}), & |\mathcal{E}| \leq n-k\\ 1, & \text{otherwise.}\end{cases}$$

From this, (3.2) follows in a straightforward manner.

Next consider the bit erasure probability. We claim that a bit $i \in \mathcal{E}$ *cannot be recovered* by an ML decoder iff $H_{\{i\}}$ is an element of the space spanned by columns of $H_{\mathcal{E}\backslash\{i\}}$. To see this we argue as follows. Write the basic equation $Hx^T = 0^T$ in the form

$$H_{\mathcal{E}\backslash\{i\}}x_{\mathcal{E}\backslash\{i\}}^T = H_{\overline{\mathcal{E}}}x_{\overline{\mathcal{E}}}^T + H_{\{i\}}x_i = s^T + H_{\{i\}}x_i.$$

Since, by assumption, x is a codeword we know that there is *at least one* choice of x_i such that this set of equations has solutions. The ML decoder will not be able to determine x_i if this equation has solutions for *both* choices of x_i. But this happens iff $H_{\{i\}}$ is contained in the column space spanned by $H_{\mathcal{E}\backslash\{i\}}$, as claimed. From (A1) we know that the probability that $H_{\mathcal{E}\backslash\{i\}}$ has rank j is equal to

$$\frac{R(|\mathcal{E}|-1, n-k, j)}{2^{(n-k)(|\mathcal{E}|-1)}}.$$

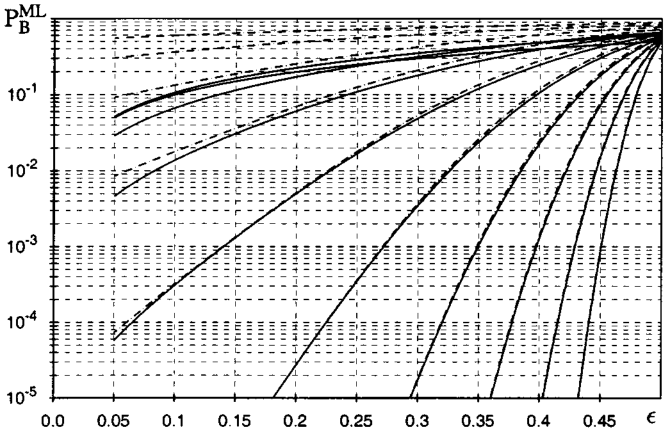

Fig. 8. $\mathbb{E}_{\mathcal{C}(n,\,k)}[\mathrm{P}_{\mathrm{b}}^{\mathrm{ML}}(H,\,\epsilon)]$ as a function of ϵ for $n = 2^i$, $i \in [10]$ (solid curves). Also shown is the union bound (dashed curves). As we can see, for increasing lengths the union bound expressions become more and more accurate.

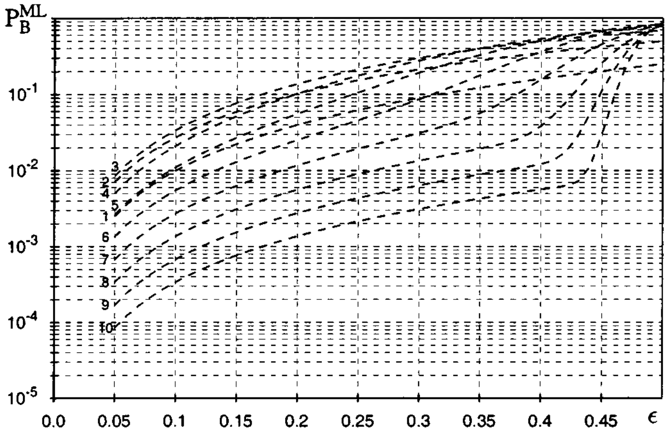

Fig. 9. Union bound on the quantity $\mathbb{E}_{\mathcal{C}(n,\,x^2,\,x^5)}[\mathrm{P}_{\mathrm{B}}^{\mathrm{ML}}(H,\,\epsilon)]$ as a function of ϵ for $n = 2^i$, $i \in [10]$.

Further, assuming that $H_{\mathcal{E}\setminus\{i\}}$ has rank j, the probability that $H_{\{i\}}$ is an element of the space spanned by the columns of $H_{\mathcal{E}\setminus\{i\}}$ is equal to 2^{j-n+k}. From these two observations (3.1) follows easily. □

Example 2: Fig. 8 shows $\mathbb{E}_{\mathcal{C}(n,k)}[\mathrm{P}_{\mathrm{b}}^{\mathrm{ML}}(H,\epsilon)]$ as a function of ϵ for $n = 2^i$, $i \in [10]$. Also shown is the union bound which is derived in Appendix B. As we can see, for increasing lengths the union bound expressions become more and more accurate.

B. LDPC Ensemble Under ML Decoding

We have so far not succeeded in deriving exact expressions for the ML performance of LPDC ensembles. From the previous section though one can see that the union bound on the ML erasure probability for the random standard random ensemble is reasonably tight except for very short lengths. Therefore, it is meaningful to derive the union bound of the ML performance of LDPC ensembles as well. This is done in Appendix B. Stronger bounds, especially away from the error floor region, can be obtained using more powerful techniques, see, e.g., [4].

Example 3: Fig. 9 shows the union bound on the quantity $\mathbb{E}_{\mathcal{C}(n,x^2,x^5)}[\mathrm{P}_{\mathrm{B}}^{\mathrm{ML}}(H,\epsilon)]$ as a function of ϵ for $n = 2^i$, $i \in [10]$.

IV. Interpretation

In comparing Fig. 7 with Fig. 9 (assuming that the shown union bound is reasonably tight) and Fig. 6 with Fig. 8, we see, at least for the ensemble $\mathcal{C}(n,\,x^2,\,x^5)$, that most of the performance loss is due to the structure of the codes themselves. Notice that for the ensemble $\mathcal{C}(n,\,x^2,\,x^5)$ the performance under iterative decoding is only slightly worse (at least in the "error floor region") than the performance under ML decoding. In particular, even under ML decoding the curves show an "error floor" region which is so characteristic of iterative coding systems. We remark that this effect is even more pronounced since we look here at block error curves. The corresponding bit error curves would show this error floor to a lesser degree.

V. Outlook

Although the exact characterization of the average bit and block erasure probabilities given in this paper are quite encouraging, much work remains to be done. We briefly gather here what we consider to be the most interesting open problems.

1) In Fig. 1 we see that the individual bit erasure curves fall into two categories. There is one curve which shows a fairly pronounced "error floor," whereas all other curves exhibit a much steeper slope. In the region where the individual curves diverge, the ensemble average is to a large degree dominated by those "bad" graphs. This suggest that one can define an *expurgated* ensemble and that the concentration of the individual behavior around the average of this expurgated ensemble holds down to much lower erasure probabilities. The question is how to find a suitable definition of such an expurgated ensemble and whether one can still find the ensemble average of the expurgated ensemble? Some progress in this direction has been made in [10].

2) The exact evaluation of

$$\mathbb{E}_{\mathcal{C}(n,\,x^2,\,x^5)}[\mathrm{P}_{\mathrm{b}}^{\mathrm{IT}}(\mathtt{G},\,\epsilon)] \quad \text{and} \quad \mathbb{E}_{\mathcal{C}(n,\,x^2,\,x^5)}[\mathrm{P}_{\mathrm{B}}^{\mathrm{IT}}(\mathtt{G},\,\epsilon)]$$

is, in general, a nontrivial task and it would be highly desirable to find simpler expressions. It is particularly frustrating that not even the simple recursion for the cycle code case seems amenable to an analytic attack. For example, if we try the obvious path employing generating functions, the resulting partial differential equation does not seem to admit an analytic solution. Simpler bounds on these quantities would also be useful.

3) Once simpler expressions for the regular case have been found, it is natural to investigate if exact expressions can also be given for the irregular case.

4) These expressions can then be used to find the *optimum* degree distribution pairs for a given length n.

5) Find exact expressions for the bit and block erasure probability for LDPC ensembles under ML decoding. Comparing then the expressions for the iterative decoding of

LDPC ensembles with the ones for the ML of LDPC ensembles and the ones for the ML of standard random ensembles it will then be possible to assess how much loss is due to the structure of the codes and how much loss is due to the suboptimum decoding. A related but simpler problem is to find the threshold for LDPC codes below which the *block* erasure probability can be made arbitrarily small. It should be interesting to see for which codes the threshold for bit and block erasure probability are different and for which they are the same. Some partial answers to the last question can be found in [10].

6) Find exact expressions for the bit and block error probability of LDPC ensembles under iterative decoding for more general channels.

7) Apply the same analysis to other ensembles, e.g., repeat–accumulate (RA) ensembles [2].

8) In this paper, we assumed that the decoder proceeds until no further progress is achieved. What is the distribution of the number of required iterations? Also, since measurements by MacKay and Kanter have indicated that the distribution of the number of required iterations have slowly decaying tails it is interesting to see how the error probabilities behave if we perform a fixed number of iterations.

Appendix A
Full Rank Matrices

Lemma A.1: Let $R(l, m, k)$ denote the number of binary matrices of dimension $l \times m$ and rank k. By symmetry

$$R(l, m, k) = R(m, l, k).$$

For $l \le m$

$$R(l, m, k) = \begin{cases} 1, & 0 = k < l \\ 2^{ml}\prod_{i=0}^{l-1}(1 - 2^{i-m}), & 0 < k = l \\ R(l-1, m, k)2^k \\ \quad + R(l-1, m, k-1)(2^m - 2^{k-1}), & \\ & 0 < k < l \\ 0, & \text{otherwise.} \end{cases} \tag{A1}$$

Proof: Clearly, there is exactly one $l \times m$ matrix of zero rank, namely, the all-zero matrix, so that $R(l, m, 0) = 1$, for $0 < l$. Next, note that

$$R(1, m, 1) = 2^m - 1$$

since any nonzero binary element of $\mathsf{GF}(2)^m$ forms a $1 \times m$ matrix of rank 1. Further by induction, any $(l-1) \times m$ matrix of rank $(l-1)$ can be extended to a $l \times m$ matrix of rank l in exactly

$$(2^m - 2^{l-1})$$

ways, and conversely, any $l \times m$ matrix of rank l can be mapped to a *unique* $(l-1) \times m$ matrix of rank $(l-1)$ by deleting the last row. It follows that

$$R(l, m, l) = R(l-1, m, l-1)(2^m - 2^{l-1}), \qquad 2 \le l \le m$$

and, therefore, that

$$R(l, m, l) = \prod_{i=0}^{l-1}(2^m - 2^i) = 2^{ml}\prod_{i=0}^{l-1}(1 - 2^{i-m}).$$

Finally, to prove the recursion we argue as follows. Consider the number of matrices of dimension $l \times m$ and rank k. Split these matrices into those matrices such that after deletion of the last row the resulting matrices of dimension $(l-1) \times m$ have rank k and those that have rank $(k-1)$. The first such group has by definition cardinality $R(l-1, m, k)$ and each element in this group can be extended to a $l \times m$ matrix of rank k in exactly 2^k distinct ways. The second group has cardinality $R(l-1, m, k-1)$ and each element in this group can be extended to a $l \times m$ matrix of rank k in exactly $(2^m - 2^{k-1})$ distinct ways. □

Appendix B
Union Bounds

It is useful to derive union bounds on the block and bit erasure probabilities of the standard random ensemble as well as for LDPC ensembles under ML decoding. We start with the standard random ensemble.

A. Random Ensembles

Lemma B.1 [Union Bound for Standard Random Ensembles Under ML Decoding]:

$$\begin{aligned} \mathbb{E}_{\mathcal{C}(n,k)}[P_{\mathrm{b}}^{\mathrm{ML}}(H, \epsilon)] \le & \sum_{e=0}^{n-k}\binom{n}{e}\epsilon^e\bar{\epsilon}^{n-e}\frac{e}{n}2^{e-n+k} \\ & + \sum_{e=n-k+1}^{n}\binom{n}{e}\epsilon^e\bar{\epsilon}^{n-e}\frac{e}{n}, \\ \mathbb{E}_{\mathcal{C}(n,k)}[P_{\mathrm{B}}^{\mathrm{ML}}(H, \epsilon)] \le & \sum_{e=0}^{n-k}\binom{n}{e}\epsilon^e\bar{\epsilon}^{n-e}2^{e-n+k} \\ & + \sum_{e=n-k+1}^{n}\binom{n}{e}\epsilon^e\bar{\epsilon}^{n-e}. \end{aligned}$$

Proof: First note that

$$\begin{aligned} &\Pr\{\mathrm{rank}(H_{\mathcal{E}}) < |\mathcal{E}|\} \\ &\quad = \Pr\{\exists x \in \mathsf{GF}(2)^{|\mathcal{E}|}\backslash\{0\}: H_{\mathcal{E}}x^T = 0^T\} \\ &\quad \le \sum_{x \in \mathsf{GF}(2)^{|\mathcal{E}|}\backslash\{0\}} \Pr\{H_{\mathcal{E}}x^T = 0^T\} \\ &\quad = \sum_{x \in \mathsf{GF}(2)^{|\mathcal{E}|}\backslash\{0\}} 2^{k-n} \\ &\quad < 2^{|\mathcal{E}|-n+k}. \end{aligned}$$

Therefore,

$$\begin{aligned} &\mathbb{E}_{\mathcal{C}(n,k)}[P_{\mathrm{B}}^{\mathrm{ML}}(H, \epsilon)] \\ &\quad = \sum_{\mathcal{E} \subseteq [n]} \Pr\{\mathcal{E}\}\Pr\{\mathrm{rank}(H_{\mathcal{E}}) < |\mathcal{E}|\} \end{aligned}$$

$$
\begin{aligned}
&= \sum_{\mathcal{E}\subseteq[n]:\,|\mathcal{E}|\le n-k} \Pr\{\mathcal{E}\}\Pr\{\mathrm{rank}(H_{\mathcal{E}}) < |\mathcal{E}|\} \\
&\quad + \sum_{\mathcal{E}\subseteq[n]:\,|\mathcal{E}|> n-k} \Pr\{\mathcal{E}\}\Pr\{\mathrm{rank}(H_{\mathcal{E}}) < |\mathcal{E}|\} \\
&< \sum_{\mathcal{E}\subseteq[n]:\,|\mathcal{E}|\le n-k} \Pr\{\mathcal{E}\}2^{|\mathcal{E}|-n+k} + \sum_{\mathcal{E}\subseteq[n]:\,|\mathcal{E}|> n-k} \Pr\{\mathcal{E}\} \\
&\le \sum_{e=0}^{n-k}\binom{n}{e}\epsilon^e\bar{\epsilon}^{n-e}2^{e-n+k} + \sum_{e=n-k+1}^{n}\binom{n}{e}\epsilon^e\bar{\epsilon}^{n-e}. \qquad \square
\end{aligned}
$$

B. LDPC Ensembles

In exactly the same manner we can derive bounds on the erasure probabilities for LDPC codes under ML decoding.

Lemma B.2 [Union Bound for LDPC Codes Under ML Decoding]:

$$
\begin{aligned}
&\mathbb{E}_{\mathcal{C}(n,\,x^{\mathtt{l}-1},\,x^{\mathtt{r}-1})}[P_{\mathrm{b}}^{\mathrm{ML}}(\mathtt{G},\,\epsilon)] \\
&\quad \le \sum_{e=0}^{n}\binom{n}{e}\epsilon^e\bar{\epsilon}^{n-e}\frac{e}{n}\ \min \\
&\qquad \cdot\left\{1,\ \sum_{w=1}^{e}\binom{e}{w}\frac{\mathrm{coef}\left(\left(\frac{(1+y)^{\mathtt{r}}+(1-y)^{\mathtt{r}}}{2}\right)^{n\frac{\mathtt{l}}{\mathtt{r}}},\ y^{w\mathtt{l}}\right)}{\binom{n\mathtt{l}}{w\mathtt{l}}}\right\}
\end{aligned}
$$

$$
\begin{aligned}
&\mathbb{E}_{\mathcal{C}(n,\,x^{\mathtt{l}-1},\,x^{\mathtt{r}-1})}[P_{\mathrm{B}}^{\mathrm{ML}}(\mathtt{G},\,\epsilon)] \\
&\quad \le \sum_{e=0}^{n}\binom{n}{e}\epsilon^e\bar{\epsilon}^{n-e}\ \min \\
&\qquad \cdot\left\{1,\ \sum_{w=1}^{e}\binom{e}{w}\frac{\mathrm{coef}\left(\left(\frac{(1+y)^{\mathtt{r}}+(1-y)^{\mathtt{r}}}{2}\right)^{n\frac{\mathtt{l}}{\mathtt{r}}},\ y^{w\mathtt{l}}\right)}{\binom{n\mathtt{l}}{w\mathtt{l}}}\right\}.
\end{aligned}
$$

Proof: We have

$$
\begin{aligned}
&\Pr\{\mathrm{rank}(H_{\mathcal{E}}) < |\mathcal{E}|\} \\
&\quad = \Pr\{\exists x\in \mathsf{GF}(2)^{|\mathcal{E}|}\backslash\{0\}\colon H_{\mathcal{E}}x^T = 0^T\} \\
&\quad \le \sum_{x\in\mathsf{GF}(2)^{|\mathcal{E}|}\backslash\{0\}} \Pr\{H_{\mathcal{E}}x^T = 0^T\} \\
&\quad = \sum_{x\in\mathsf{GF}(2)^{|\mathcal{E}|}\backslash\{0\}} \frac{\mathrm{coef}\left(\left(\frac{(1+y)^{\mathtt{r}}+(1-y)^{\mathtt{r}}}{2}\right)^{n\frac{\mathtt{l}}{\mathtt{r}}},\ y^{w(x)\mathtt{l}}\right)}{\binom{n\mathtt{l}}{w(x)\mathtt{l}}} \\
&\quad = \sum_{w=1}^{|\mathcal{E}|}\binom{|\mathcal{E}|}{w}\frac{\mathrm{coef}\left(\left(\frac{(1+y)^{\mathtt{r}}+(1-y)^{\mathtt{r}}}{2}\right)^{n\frac{\mathtt{l}}{\mathtt{r}}},\ y^{w\mathtt{l}}\right)}{\binom{n\mathtt{l}}{w\mathtt{l}}}
\end{aligned}
$$

where $w(x)$ denotes the weight of x, from which the block erasure probability follows in a straightforward manner. □

Acknowledgment

The authors wish to thank Igal Sason and the reviewers for their many helpful comments on an earlier draft of this paper.

References

[1] L. Bazzi, T. Richardson, and R. Urbanke, "Exact thresholds and optimal codes for the binary symmetric channel and Gallager's decoding algorithm A," *IEEE Trans. Inform. Theory*, 1999, to be published.

[2] D. Divsalar, H. Jin, and R. McEliece, "Coding theorems for "turbo-like" codes," in *Proc. 1998 Allerton Conf.*, 1998, p. 210.

[3] R. Gallager, "Low-density parity-check codes," *IRE Trans. Inform. Theory*, vol. IT-8, pp. 21–28, Jan. 1962.

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

[5] M. Luby, M. Mitzenmacher, A. Shokrollahi, and D. Spielman, "Analysis of low density codes and improved designs using irregular graphs," in *Proc. 30th Annu. ACM Symp. Theory of Computing*, 1998, pp. 249–258.

[6] ——, "Improved low-density parity-check codes using irregular graphs and belief propagation," in *Proc. 1998 IEEE Int. Symp. Information Theory*, 1998, p. 117.

[7] M. Luby, M. Mitzenmacher, A. Shokrollahi, D. Spielman, and V. Stemann, "Practical loss-resilient codes," in *Proc. 29th Annu. ACM Symp. Theory of Computing*, 1997, pp. 150–159.

[8] D. J. C. MacKay, "Good error correcting codes based on very sparse matrices," *IEEE Trans. Inform. Theory*, vol. 45, pp. 399–431, Mar. 1999.

[9] T. Richardson, A. Shokrollahi, and R. Urbanke, "Design of capacity approaching irregular low-density parity check codes," *IEEE Trans. Inform. Theory*, vol. 47, pp. 619–637, Feb. 2001.

[10] ——, "Error floor analysis of various low-density parity-check ensembles for the binary erasure channel," submitted to IEEE Int. Symp. Information Theory, Lausanne, 2002.

[11] T. Richardson and R. Urbanke, "The capacity of low-density parity check codes under message-passing decoding," *IEEE Trans. Inform. Theory*, vol. 47, pp. 599–618, Feb. 2001.

[12] ——, "Efficient encoding of low density parity check codes," *IEEE Trans. Inform. Theory*, vol. 47, pp. 638–656, Feb. 2001.

[13] A. Shokrollahi, "New sequences of linear time erasure codes approaching the channel capacity," in *Proc. 13th Conf. Applied Algebra, Error Correcting Codes, and Cryptography (Lecture Notes in Computer Science)*. New York: Springer Verlag, 1999, pp. 65–76.

[14] ——, "Capacity-achieving sequences," in *Codes, Systems, and Graphical Models*, B. Marcus and J. Rosenthal, Eds. New York: Springer-Verlag, 2000, vol. 123, IMA Volumes in Mathematics and its Applications, pp. 153–166.

[15] A. Shokrollahi and R. Storn, "Design of efficient erasure codes with differential evolution," in *Proc. Int. Symp. Information Theory*, Sorrento, 2000.

[16] J. Zhang and A. Orlitsky, "Finite length analysis of LDPC codes with large left degrees," submitted to IEEE Int. Symp. Information Theory, Lausanne, Switzerland, 2002.