On the Block Error Probability of LP Decoding of LDPC Codes

Ralf Koetter
CSL and Dept. of ECE
University of Illinois at Urbana-Champaign
Urbana, IL 61801, USA
koetter@uiuc.edu

Pascal O. Vontobel
Dept. of EECS
Massachusetts Institute of Technology
Cambridge, MA 02139, USA
pascal.vontobel@ieee.org

***Abstract*— In his thesis, Wiberg showed the existence of thresholds for families of regular low-density parity-check codes under min-sum algorithm decoding. He also derived analytic bounds on these thresholds. In this paper, we formulate similar results for linear programming decoding of regular low-density parity-check codes.**

I. Introduction

The goal of this paper is to shed some light on the connection between min-sum algorithm (MSA) decoding and the formulation of decoding as a linear program. In particular, we address the problem of bounding the performance of linear programming (LP) decoding with respect to word error rate. The bounds reflect similar analytic bounds for MSA decoding of low-density parity-check (LDPC) codes due to Wiberg [1] and establish the existence of an SNR threshold for LP decoding. While highly efficient and structured computer-based evaluation techniques, such as density evolution (see e.g. [2], [3], [4], [5], [6]), provide excellent bounds on the performance of iterative decoding techniques, to the best of our knowledge, the best analytic performance bound in the case of MSA decoding is still the bound given by Wiberg in his thesis based on the weight distribution of a tree-like neighborhood of a vertex in a graph. A similar bound was also derived by Lentmaier et al. [5]. We derive the equivalent bound for LP decoding of regular LDPC codes.

II. Notation and Basics

In this paper we are interested in binary LDPC codes where a binary LDPC code $\mathcal{C}$ of length n is defined as the null-space of a sparse binary parity-check matrix $\mathbf{H}$, i.e. $\mathcal{C} \triangleq \{\mathbf{x} \in \mathbb{F}_2^n \mid \mathbf{H}\mathbf{x}^{\mathsf{T}} = \mathbf{0}^{\mathsf{T}}\}$. In particular, we focus on the case of regular codes: an LDPC code $\mathcal{C}$ is called (J, K)-regular if each column of $\mathbf{H}$ has Hamming weight J and each row has Hamming weight K. The rate of a (J, K)-regular code is lower bounded by $1-J/K$. To an $M \times N$ parity-check matrix $\mathbf{H}$ we can naturally associate a bipartite graph, the so-called Tanner graph $\mathsf{T}(\mathbf{H})$. This graph contains two classes of nodes: variable nodes $\mathcal{V}_{\mathrm{v}}$ and check nodes $\mathcal{V}_{\mathrm{c}}$. Both variable nodes and check nodes are identified with subsets of the integers. Variable nodes are denoted as $\mathcal{V}_{\mathrm{v}} \triangleq \{0, 1, \ldots, N-1\}$ and check nodes are denoted as $\mathcal{V}_{\mathrm{c}} \triangleq \{0, 1, \ldots, M-1\}$. Whenever we want to express that an integer belongs to the set of variable nodes we write $i \in \mathcal{V}_{\mathrm{v}}$; similarly, when an integer belongs to the set of check nodes we write $j \in \mathcal{V}_{\mathrm{c}}$. The Tanner graph $\mathsf{T}(\mathbf{H})$ contains an edge (i, j) between node $i \in \mathcal{V}_{\mathrm{v}}$ and $j \in \mathcal{V}_{\mathrm{c}}$ if and only if the entry $h_{i,j}$ is non-zero. The set of neighbors of a node $i \in \mathcal{V}_{\mathrm{v}}$ is denoted as $\partial(i)$; similarly, the set of neighbors of a node $j \in \mathcal{V}_{\mathrm{c}}$ is denoted as $\partial(j)$. In the following, $\mathcal{E} \triangleq \{(i,j) \in \mathcal{V}_{\mathrm{v}} \times \mathcal{V}_{\mathrm{c}} \mid i \in \mathcal{V}_{\mathrm{v}},\ j \in \partial(i)\} = \{(i,j) \in \mathcal{V}_{\mathrm{v}} \times \mathcal{V}_{\mathrm{c}} \mid j \in \mathcal{V}_{\mathrm{c}},\ i \in \partial(j)\}$ will be the set of edges in the Tanner graph $\mathsf{T}(\mathbf{H})$. The convex hull of a set $\mathcal{A} \subseteq \mathbb{R}^n$ is denoted by $\operatorname{conv}(\mathcal{A})$. If $\mathcal{A}$ is a subset of $\mathbb{F}_2^n$ then $\operatorname{conv}(\mathcal{A})$ denotes the convex hull of the set $\mathcal{A}$ after $\mathcal{A}$ has been canonically embedded in $\mathbb{R}^n$. The inner product between vectors $\mathbf{x}$ and $\mathbf{y}$ is denoted as $\langle \mathbf{x}, \mathbf{y} \rangle = \sum_l x_l y_l$. Finally, we define the set of all binary vectors of length K and even weight as $\mathcal{B}^{(K)}$.

In the rest of this paper we assume that the all-zeros word was transmitted — an assumption without any essential loss of generality because we only consider binary linear codes that are used for data transmission over a binary-input output-symmetric channel. Given a received vector $\mathbf{y}$ we define the vector $\boldsymbol{\lambda} \triangleq (\lambda_0, \lambda_1, \ldots, \lambda_{N-1})$ of log-likelihood ratios by

$$\lambda_i = \log\left(\frac{P_{Y|X}(y_i|0)}{P_{Y|X}(y_i|1)}\right).$$

III. LP Decoding

Maximum likelihood (ML) decoding may be cast as a linear program once we have translated the problem into $\mathbb{R}^N$. To this end we embed the code into $\mathbb{R}^N$ by straightforward identification of $\mathbb{F}_2 = \{0, 1\}$ with $\{0, 1\} \subset \mathbb{R}$. In other words, a code $\mathcal{C}$ is identified with a subset of $\{0, 1\}^N \subset \mathbb{R}^N$.

Maximum Likelihood Decoding
Minimize: $\langle \boldsymbol{\lambda}, \mathbf{x} \rangle$
Subject to: $\mathbf{x} \in \operatorname{conv}(\mathcal{C})$

This description is usually not practical since the polytope $\operatorname{conv}(\mathcal{C})$ is typically very hard to describe by hyperplanes (or as a convex combination of points). Given a parity-check matrix $\mathbf{H}$, the linear program is relaxed to [7], [8]

LP Decoding

Minimize: $\langle \boldsymbol{\lambda}, \mathbf{x} \rangle$

Subject to: $\mathbf{x} \in \mathcal{P}(\mathbf{H})$

Here, $\mathcal{P}(\mathbf{H})$ is the so-called fundamental polytope [7], [8], [9], [10] which is defined as

$$\mathcal{P}(\mathbf{H}) \triangleq \bigcap_{j=0}^{M-1} \operatorname{conv}(\mathcal{C}_j),$$

where

$$\mathcal{C}_j \triangleq \mathcal{C}_j(\mathbf{H}) \triangleq \left\{ \mathbf{x} \in \mathbb{F}_2^n \mid \mathbf{h}_j \mathbf{x}^{\mathsf{T}} = 0 \ (\mathrm{mod}\ 2) \right\},$$

where $\mathbf{h}_j$ is the j-th row of $\mathbf{H}$.

Since $\mathbf{0}$ is always a feasible point, i.e. $\mathbf{0} \in \mathcal{P}(\mathbf{H})$ holds, zero is an upper bound on the value of the LP in LP decoding. In fact, we can turn this statement around by saying that whenever the value of the linear program equals zero then the all-zeros codeword will be among the solutions to the LP. Thus, motivated by the assumption that the all-zeros codeword was transmitted, we focus our attention on showing that, under suitable conditions, the value of the LP is zero which implies that the all-zeros codeword will be found as a solution. For simplicity we only consider channels where the channel output is a continuous random variable. In this case a zero value of the LP implies that the zero word is the unique solution with probability one. The main idea now is to show that the value of the dual linear program is zero. This technique, dubbed "dual witness" by Feldman et al. in [11] will then imply the correct decoding.

First, however, we need to establish the dual linear program. To this end, for each $(i, j) \in \mathcal{E}$, we associate the variable $\tau_{i,j}$ with the edge between variable node i and check node j in the Tanner graph $\mathsf{T}(\mathbf{H})$. In other words, we have a variable $\tau_{i,j}$ if and only if the entry $h_{i,j}$ is non-zero. For each $j \in \mathcal{V}_{\mathrm{c}}$ we define the vector $\boldsymbol{\tau}_j$ that collects all the variables $\{\tau_{i,j}\}_{i \in \partial(j)}$. Also, for each $j \in \mathcal{V}_{\mathrm{c}}$, we associate the variable θ_j with the check node j. We have[1]

Dual LP

Maximize: $\sum_{j=0}^{M-1} \theta_j$

Subject to: $\theta_j \leq \langle \mathbf{x}, \boldsymbol{\tau}_j \rangle \qquad \forall\ j \in \mathcal{V}_{\mathrm{c}}, \quad \forall \mathbf{x} \in \mathcal{B}^{(K)}$

$\sum_{j \in \partial(i)} \tau_{i,j} = \lambda_i \qquad \forall\ i \in\ \mathcal{V}_{\mathrm{v}}$

The dual program has a number of nice interpretations. Any θ_j is bounded from above by zero and can only equal zero if the vector $\boldsymbol{\tau}_j$ has minimal correlation with the all-zeros codeword.[2] Thus the dual program will only get a zero value if we find an assignment to $\tau_{i,j}$ such that the local all-zeros words are among the "best" words for all j. We are constraint in setting the $\tau_{i,j}$-values by the second equality constraint.

IV. MSA Decoding

While MSA decoding is not the focus of interest in this paper, it turns out that the MSA lies at the core of the proof technique that we will use. The MSA is an algorithm that is being run until a predetermined criterion is reached. With each edge in the graph we associate two messages: one message is going towards the check-node and one is directed towards the variable node. Let the two messages be denoted by $\mu_{i,j}$ and $\nu_{i,j}$, respectively, where, as in the case of the single variable $\tau_{i,j}$ in the section above, variables are only defined if the entry $h_{i,j}$ is non-zero. The update rules of MSA are then

Min-Sum Algorithm (MSA)

Initialize all variables $\nu_{i,j}$ to zero.

1) For all $(i, j) \in \mathcal{E}$, let

$$\mu_{i,j} := \lambda_i + \sum_{j' \in \partial(i) \setminus \{j\}} \nu_{j',i}.$$

2) For all $(i, j) \in \mathcal{E}$, let

$$\nu_{i,j} := \left(\prod_{i' \in \partial(j) \setminus \{i\}} \operatorname{sign}(\mu_{j,i'}) \right) \cdot \min \left\{ |\mu_{j,i'}| \ : \ i' \in \partial(j) \setminus \{i\} \right\}.$$

Rather than the quantity $\nu_{i,j}$ we will consider its negative value. Moreover, we keep track of the messages that were sent by message numbers in the superscript. Thus we modify the MSA update equations as

Modified Min-Sum Algorithm (modified MSA)

Initialize all variables $\nu_{i,j}^{(1)}$ to zero.

1) For all $(i, j) \in \mathcal{E}$, let

$$\mu_{i,j}^{(s)} := \lambda_i - \sum_{j' \in \partial(i) \setminus \{j\}} \nu_{j',i}^{(s)}.$$

2) For all $(i, j) \in \mathcal{E}$, let

$$\nu_{i,j}^{(s+1)} := - \left(\prod_{i' \in \partial(j) \setminus \{i\}} \operatorname{sign}(\mu_{j,i'}^{(s)}) \right) \cdot \min \left\{ \left| \mu_{j,i'}^{(s)} \right| \ : \ i' \in \partial(j) \setminus \{i\} \right\}.$$

Clearly, the sign change leaves the algorithmic update steps essentially unchanged. (Note that e.g. when all $\{\mu_{i,j}^{(s)}\}_{i \in \partial(j)}$ are non-negative then all $\{\nu_{i,j}^{(s)}\}_{i \in \partial(i)}$ will be non-positive.) Still, we may e.g. write $\mu_{i,j}^{(s)} + \sum_{j' \in \partial(i) \setminus \{j\}} \nu_{j',i}^{(s)} = \lambda_i$ which more closely reflects the structure of the dual program above.

We will need the notion of a computation tree (CT) [1]. We can distinguish two types of CTs, rooted either at a variable node or at a check node. Our CTs will be rooted at check nodes which is more natural when dealing with the dual program. A CT of depth L consists of all nodes in the universal cover of the Tanner graph that are reachable in $2L - 1$ steps. In

[1] In the formal dual program the equality constraint $\sum_{j \in \partial(i)} \tau_{i,j} = \lambda_i$ is an inequality ($\leq$). However, there always exists a maximizing assignment of dual variables that satisfies this conditions with equality.

[2] In a generalized LDPC code setting, the local code $\mathcal{B}^{(K)}$ would have to be replaced by the corresponding code.

particular, we will most of the time assume that the leaves in the CT are variable nodes.

Assume we have run the MSA for L iterations, corresponding to a CT of depth L. For the moment let us also assume that the underlying graph has girth larger than $4L$. Based on the iterations of the MSA and fixed CT root node $j_0 \in \mathcal{V}_{\mathrm{c}}$ we can assign values to the dual variables in the following way.

Was assign values to $\tau_{i,j}$ according to the distance of the edge (i,j) to the root node of the CT. So, if (i,j) is at distance $2\ell+1$ from the root node j_0 then $\tau_{i,j}$ is assigned the value $\mu_{i,j}^{(L-\ell)}$ and if (i,j) is at distance $2\ell+2$ from the root node j_0 then $\tau_{i,j}$ is assigned the value $\nu_{i,j}^{(L-\ell)}$.[3] Let us denote this assignment to variables $\tau_{i,j}$ as $\boldsymbol{\tau}(j_0, L)$[4]. Note that the assignment $\boldsymbol{\tau}(j_0, L)$ does not satisfy the constraints of the dual linear program, i.e. itself it is not dual feasible. In particular, any edge of distance more than $2L$ from the root is assigned the value 0 and hence at any variable node i at distance more than $2L$ from the root we do not satisfy the constraint

$$\sum_{j \in \partial(i)} \tau_{i,j} = \lambda_i,$$

unless λ_i happens to be 0. However, we have the following lemma.

Lemma 1: For each $j_0 \in \mathcal{V}_{\mathrm{c}}$ let an assignment $\boldsymbol{\tau}(j_0, L)$ be given based on L iterations of the MSA. The sum

$$\boldsymbol{\tau}(L) \triangleq \sum_{j_0 \in \mathcal{V}_{\mathrm{c}}} \boldsymbol{\tau}(j_0, L)$$

is a multiple of a dual feasible point. More precisely, for the number $T(L) \triangleq \sum_{\ell=1}^{L} J\big[(K-1)(J-1)\big]^{(\ell-1)}$ the vector

$$\frac{1}{T(L)} \boldsymbol{\tau}(L)$$

is a dual feasible point.

Proof: Each variable node $i \in \mathcal{V}_{\mathrm{v}}$ is part of $\sum_{\ell=1}^{L} J\big[(K-1)(J-1)\big]^{(\ell-1)}$ CTs for different root nodes j_0 and so one can verify that we must have $\tau_{i,j}(L) = \sum_{j_0 \in \mathcal{V}_{\mathrm{c}}} \tau_{i,j}(j_0, L) = \sum_{\ell=1}^{L} J\big[(K-1)(J-1)\big]^{(\ell-1)} \lambda_i$. Using the abbreviation $T(L) \triangleq \sum_{\ell=1}^{L} J\big[(K-1)(J-1)\big]^{\ell-1}$ we see that

$$\frac{1}{T(L)} \boldsymbol{\tau}(L)$$

is a dual feasible point. □

The above lemma gives a structured way to derive dual feasible points for LP decoding from the messages passed during the operation of the MSA. However, these points are not very good since the overall assignment $\boldsymbol{\tau}(L)$ is again dominated by the leaves of the CT with all the pertaining problems. The problem becomes obvious when we write out the assignment $\boldsymbol{\tau}(L)$ as a function of the MSA messages directly. If we perform L steps of iterative decoding, for any edge $(i,j) \in \mathcal{E}$ we can write

$$\begin{aligned}\tau_{i,j}(L) = \mu_{i,j}^{(L)} &+ (J-1)\left(\nu_{i,j}^{(L)} + (K-1)\mu_{i,j}^{(L-1)}\right) \\ &+ (J-1)^2(K-1)\left(\nu_{i,j}^{(L-1)} + (K-1)\mu_{i,j}^{(L-2)}\right) \\ &+ \cdots \ .\end{aligned}$$

Written in form of a telescoping sum we get

$$\begin{aligned}\tau_{i,j}(L) = \mu_{i,j}^{(L)} + (J-1)\Bigg(\nu_{i,j}^{(L)} + (K-1)\Bigg(\mu_{i,j}^{(L-1)} + \\ (J-1)\Big(\nu_{i,j}^{(L-1)} + (K-1)\big(\mu_{i,j}^{(L-2)} + \cdots\big)\Big)\Bigg)\Bigg).\end{aligned}$$

While the above sums show that the dual feasible point can be easily computed alongside the MSA recursions it also shows the problem that messages $\mu_{i,j}^{(\ell)}$ and $\nu_{i,j}^{(\ell)}$ are weighted exponentially more for small values of ℓ.

We will have to attenuate the influence of the leaves in the CTs in order to make interesting statements. To this end, let $\boldsymbol{\alpha}$ be a vector with positive entries of length L and let a generalized assignment $\boldsymbol{\tau}(j_0, L, \boldsymbol{\alpha})$ to dual variables be derived from $\boldsymbol{\tau}(j_0, L)$ by multiplying the message on each edge at distance $2\ell+1$ or $2\ell+2$ by α_ℓ.[5] In other words, values assigned to edges at distance three or four from the root node are multiplied with α_1, values at distance five and six are multiplied with α_2 etc. Again we can form the multiple of a dual feasible point as is shown in the next lemma.

Lemma 2: For each $j_0 \in \mathcal{V}_{\mathrm{c}}$ let an assignment $\boldsymbol{\tau}(j_0, L)$ be given based on L iterations of the MSA. The sum

$$\boldsymbol{\tau}(L, \boldsymbol{\alpha}) \triangleq \sum_{j_0 \in \mathcal{V}_{\mathrm{c}}} \boldsymbol{\tau}(j_0, L, \boldsymbol{\alpha})$$

is a multiple of a dual feasible point.

Proof: Each variable node $i \in \mathcal{V}_{\mathrm{v}}$ is part of $\sum_{\ell=1}^{L} J\big[(K-1)(J-1)\big]^{(\ell-1)}$ CTs for different root nodes j_0. Because all edges incident to a variable node are attenuated in the same way, one can verify that we must have $\tau_{i,j}(L, \boldsymbol{\alpha}) = \sum_{j_0 \in \mathcal{V}_{\mathrm{c}}} \tau_{i,j}(j_0, L, \boldsymbol{\alpha}) = \sum_{\ell=1}^{L} \alpha_{\ell-1} J\big[(K-1)(J-1)\big]^{(\ell-1)} \lambda_i$. Using the abbreviation $T(L) \triangleq \sum_{\ell=1}^{L} \alpha_{\ell-1} J\big[(K-1)(J-1)\big]^{\ell-1}$ we see that

$$\frac{1}{T(L)} \boldsymbol{\tau}(L)$$

is a dual feasible point. □

Optimizing the vector $\boldsymbol{\alpha}$ gives us some freedom and we want to choose the vector $\boldsymbol{\alpha}$ appropriately. First we have to learn more about the dual feasible point that we construct in this way. While we kept the feasibility of an assignment $\boldsymbol{\tau}(L, \boldsymbol{\alpha})$ by identically scaling the values $\tau_{i,j}$ that are adjacent to a variable node in a CT, we scale values $\tau_{i,j}$ that are adjacent to check nodes differently. Given a vector $\boldsymbol{\alpha}$, the dual feasible

[3] Edges incident to the root are said to be at distance one. If the distance of the edge (i,j) to the root j_0 is larger than $2L$ then $\tau_{i,j} \triangleq 0$.

[4] The j_0 indicates that the assignment is based on the CT rooted at node j_0.

[5] An edge that is incident to a node j is said to be at distance one from j; α_0 is set to one.

point may be easily computed together with the messages of the MSA. To this end define a vector $\boldsymbol{\beta}$ with components $\beta_\ell \triangleq \frac{\alpha_\ell}{\alpha_{\ell-1}}$. Writing again the dual variable $\tau_{i,j}(L,\boldsymbol{\alpha})$ as functions of $\mu_{i,j}^{(\ell)}$ and $\nu_{i,j}^{(\ell)}$ we get

$$\begin{aligned}
&\tau_{i,j}(L,\boldsymbol{\alpha}) \\
&= \mu_{i,j}^{(L)} + (J-1)\left(\nu_{i,j}^{(L)} + (K-1)\alpha_1\mu_{i,j}^{(L-1)}\right) \\
&\quad + (J-1)^2(K-1)\left(\alpha_1\nu_{i,j}^{(L-1)} + (K-1)\alpha_2\mu_{i,j}^{(L-2)}\right) \\
&\quad + \cdots .
\end{aligned}$$

Written in form of a telescoping sum we obtain

$$\begin{aligned}
&\tau_{i,j}(L,\boldsymbol{\alpha}) \\
&= \mu_{i,j}^{(L)} + (J-1)\Bigg(\nu_{i,j}^{(L)} + \beta_1(K-1)\Bigg(\mu_{i,j}^{(L-1)} + \\
&\quad (J-1)\Big(\nu_{i,j}^{(L-1)} + \beta_2(K-1)\big(\mu_{i,j}^{(L-2)} + \cdots\big)\Big)\Bigg)\Bigg).
\end{aligned}$$

A particularly interesting choice for β_ℓ is $\beta_\ell \triangleq \frac{1}{K-1}$. The main reason for this choice is given in the following lemma.

Lemma 3: Let $K > 2$ and fix some $j \in \mathcal{V}_{\mathrm{c}}$. Assume the MSA yields messages where $\mu_{i,j}^{(\ell)}$ is positive for all $i \in \partial(j)$ for some ℓ. The inner product

$$\sum_{i\in\partial(j)} b_i \left(\mu_{i,j}^{(\ell)} + \nu_{i,j}^{(\ell+1)}\right)$$

is non-negative for all $\mathbf{b} \in \mathcal{B}^{(K)}$, in particular it is positive for all $\mathbf{b} \in \mathcal{B}^{(K)} \setminus \{\mathbf{0}\}$.[6]

Proof: Recall that $\nu_{i,j}^{(\ell)}$ is negative for all $(i,j) \in \mathcal{E}$ (this is in line with the modification of the MSA). One can easily verify the following fact about the vector containing $\mu_{i,j}^{(\ell)} + \nu_{i,j}^{(\ell+1)}$ for all $i \in \partial(j)$: there is only one negative entry and the absolute value of this entry matches the absolute value of the smallest positive entry. The statement follows. □

With the choice of $\alpha_i \triangleq (K-1)^{-i}$, which results in $\beta_i = \frac{1}{K-1}$, we get the following expression for the dual feasible point

$$\begin{aligned}
\tau_{i,j}(L,\boldsymbol{\alpha}) = \mu_{i,j}^{(L)} + (J-1)\Bigg(&\nu_{i,j}^{(L)} + \mu_{i,j}^{(L-1)} \\
&+ (J-1)\Big(\nu_{i,j}^{(L-1)} + \mu_{i,j}^{(L-2)} + (J-1)(\cdots)\Big)\Bigg)
\end{aligned}$$

or

$$\begin{aligned}
\tau_{i,j}(L,\boldsymbol{\alpha}) = \mu_{i,j}^{(L)} &+ (J-1)\left(\nu_{i,j}^{(L)} + \mu_{i,j}^{(L-1)}\right) \\
&+ (J-1)^2\left(\nu_{i,j}^{(L-1)} + \mu_{i,j}^{(L-2)}\right) \\
&+ \cdots + (J-1)^{L-1}\left(\nu_{i,j}^{(2)} + \mu_{i,j}^{(1)}\right).
\end{aligned}$$

We are still in a situation where $\mu_{i,j}^{(1)}$ is weighted by a factor that grows exponentially fast in L. However, we note that, once the MSA has converged, $\mu_{i,j}^{(\ell)}$ also grows exponentially fast in ℓ and this offsets, to some extend, the exponential weighing of $\mu_{i,j}^{(1)}$. In order to exploit this fact more systematically we initialize the MSA's check to variable messages $\nu_{i,j}^{(1)}$, $(i,j) \in \mathcal{E}$, with $-U$, where U is a large enough positive number. With this initialization we can guarantee (for $K > 2$) for all $(i,j) \in \mathcal{E}$ that the value of $\mu_{i,j}^{(\ell)}$ is strictly positive.[7] Thus we can apply Lemma 3. It remains to offset the choice $\nu_{i,j}^{(1)}$ with $\mu_{i,j}^{(L)}$.

To this end we consider a CT of depth L rooted at check node j_0. Consider the event A_{j_0} that the all-zeros word on this CT is more likely than any word that corresponds to a local nonzero word assigned to the root node. [8].

Lemma 4: Let $K > 2$ and assume the event A_{j_0} is true. Moreover, assume that we initialize the MSA with check to variable messages $\nu_{i,j}^{(1)} = -U$, $(i,j) \in \mathcal{E}$, for a large enough number U. The inner product

$$\sum_{i\in\partial(j)} b_i \left(\mu_{i,j}^{(L)} + (J-1)^L \nu_{i,j}^{(1)}\right)$$

is non-negative for all $\mathbf{b} \in \mathcal{B}^{(K)}$, in particular it is non-negative for all $\mathbf{b} \in \mathcal{B}^{(K)} \setminus \{\mathbf{0}\}$.

Proof: We exploit the fact that summaries sent by the MSA can be identified with cost differences of log-likelihood ratios. Consider a message $\mu_{i,j_0}^{(L)}$ on edge (i,j_0). This message may be written as $\mu_{i,j_0}^{(L)} = \rho_i - (J-1)^L\nu_{i,j_0}^{(1)}$ for some ρ_i. Since the MSA propagates cost summaries along edges, we can interpret ρ_i as the summary of the cost due to the λ_i inside the subtree that emerges along the edge (i,j_0). Similarly, $(J-1)^L\nu_{i,j_0}^{(1)}$ is the cost contributed by the leaf nodes of this sub-tree. Here we use the fact that the minimal codeword which accounts for a one-assignment in edge (i,j_0) contains exactly $(J-1)^L$ leaf nodes with a one-assignment. But then the vector $(\mu_{1,j_0}^{(L)}, \mu_{2,j_0}^{(L)}, \ldots, \mu_{|\partial(j_0)|,j_0}^{(L)}) + (J-1)^L(\nu_{1,j_0}^{(1)}, \nu_{2,j_0}^{(1)}, \ldots, \nu_{|\partial(j_0)|,j_0}^{(1)})$ equals the vector $\boldsymbol{\rho} \triangleq (\rho_1, \rho_2, \ldots, \rho_{|\partial(j_0)|})$. The event A_{j_0} is true only if the inner product $\langle \boldsymbol{\rho}, \mathbf{b} \rangle$ is positive for all $\mathbf{b} \in \mathcal{B}^{(K)} \setminus \{\mathbf{0}\}$. Hence event A_{j_0} implies the claim of the lemma. □

Let $\boldsymbol{\tau}^*$ be the averaged assignment to the dual variables obtained from the MSA messages with $\nu_{i,j}^{(1)}$ set to $-U$. Lemmas 3 and 4 imply that the sum,

$$\sum_{i\in\partial(j)} b_i \tau_{i,j_0}^*$$

has a non-negative value for any $\mathbf{b} \in \mathcal{B}^{(K)}$, and, in particular, the value equals zero for $\mathbf{b} = \mathbf{0}$. It follows that θ_{j_0} in the dual LP can be chosen as zero.

For each check node j for which event A_j is true we can be sure that the correlation of any codeword in $\mathcal{B}^{(K)}$ with $\boldsymbol{\tau}_j^*$ is non-negative. If we can be sure that the event A_j is true for all check nodes we would, thus, have exhibited a dual witness for the optimality of the all-zeros codeword. We have to estimate the probability of the event A_j and set it in relation to the

[6] We assume that the indices of $\mathbf{b}$ are given by $\partial(j)$.

[7] We may choose as any number greater than $|\min(\lambda_i)/(J-2)|$.

[8] Event A_{j_0} is defined on the CT without the change in initialization

number of checks in the graph $\mathsf{T}(\mathbf{H})$. In order to estimate the latter we employ a result by Gallager [12] that guarantees the existence of (J, K)-regular graphs in which we can conduct L steps of MSA decoding without closing any cycles provided that L satisfies

$$L \leq \frac{\log(N)}{2\log((J-1)(K-1))} - \kappa \tag{1}$$

where the term κ in this expression is independent of N.

Finally, we can estimate the probability of the event A_j from the known weight distribution of the code on the CT provided the underlying graph has girth at least $4L$. The minimal codewords have weight $2(1+(J-1)+(J-1)^2+\cdots(J-1)^{L-1})$ and there are a total of

$$\begin{aligned} &\frac{K(K-1)}{2} \cdot (K-1)^{2(J-1)} \cdot (K-1)^{2(J-1)^2} \\ &\quad \cdots (K-1)^{2(J-1)^{L-1}} \\ &= K/2(K-1)^{2(1+(J-1)+(J-1)^2+\ldots(J-1)^{L-1})} \end{aligned}$$

minimal-tree codewords. Based on a union bound we thus get an expression

$$P(A_j) < \frac{K}{2}\big((K-1)\gamma\big)^{2^{\frac{(J-1)^L-1}{J-2}}} \tag{2}$$

which means that $P(A_j)$ decreases doubly exponentially in L if the Bhattacharyya parameter γ satisfies $\gamma < \frac{1}{K-1}$.

Thus we have proved the following theorem:

Theorem 5: Let a sequence of (J, K)-regular LDPC codes be given that satisfies equation (1). Under LP decoding this sequence achieves an arbitrarily small probability of error on any memoryless channel for which the Bhattacharyya parameter γ satisfies $\gamma \leq \frac{1}{K-1}$. For such a channel the word error probability P_W decreases as

$$P_W < \eta_1 2^{-\eta_2 N^{\frac{\log(J-1)}{2\log((J-1)(K-1))}}}$$

for some positive parameters η_1 and η_2.

Proof: Most of the proof is contained in the arguments leading up to the theorem. In order to see the explicit form of the word error rate we employ a union bound for the $N\frac{J}{K}$ check nodes combining (1) and (2). We find that the word error rate is bounded by

$$P_W < \frac{NJ}{2}\big((K-1)\gamma\big)^{2^{\frac{(J-1)^{\frac{\log(N)}{2\log((J-1)(K-1))}-\kappa}-1}{J-2}}},$$

where κ does not depend on N. The statement of the theorem is obtained by simplifying this expression. □

We conclude this paper with an intriguing observation concerning the AWGN channel. In [10] it is proved that no (J, K)-regular LDPC code can achieve an error probability behavior better than $P_W \geq \eta_3 2^{-\eta_4 N^{\frac{2\log(J-1)}{\log((J-1)(K-1))}}}$ for constants η_3 and η_4 that are independent on N. The result of the theorem thus shows that there exist sequences of LDPC codes whose error probability behavior under LP decoding is boxed in as a function of N between:

$$\eta_3 2^{-\eta_4 N^{\frac{2\log(J-1)}{\log((J-1)(K-1))}}} \leq P_w \leq \eta_1 2^{-\eta_2 N^{\frac{\log(J-1)}{2\log((J-1)(K-1))}}}$$

References

[1] N. Wiberg, *Codes and Decoding on General Graphs*. PhD thesis, Linköping University, Sweden, 1996.

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

[3] M. G. Luby, M. Mitzenmacher, and M. A. Shokrollahi, "Analysis of random processes via and-or tree evaluation," in *Proc. 9th Annual ACM-SIAM Symp. on Discrete Algorithms*, pp. 364–373, 1998.

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

[5] M. Lentmaier, D. V. Truhachev, D. J. Costello, Jr., and K. Zigangirov, "On the block error probability of iteratively decoded LDPC codes," in *5th ITG Conference on Source and Channel Coding*, (Erlangen, Germany), Jan. 14-16 2004.

[6] H. Jin and T. Richardson, "Block error iterative decoding capacity for LDPC codes," in *Proc. IEEE Intern. Symp. on Inform. Theory*, (Adelaide, Australia), pp. 52 – 56, Sep. 4–9 2005.

[7] J. Feldman, *Decoding Error-Correcting Codes via Linear Programming*. PhD thesis, Massachusetts Institute of Technology, Cambridge, MA, 2003. Available online under `http://www.columbia.edu/~jf2189/pubs.html`.

[8] J. Feldman, M. J. Wainwright, and D. R. Karger, "Using linear programming to decode binary linear codes," *IEEE Trans. on Inform. Theory*, vol. IT–51, no. 3, pp. 954–972, 2005.

[9] R. Koetter and P. O. Vontobel, "Graph covers and iterative decoding of finite-length codes," in *Proc. 3rd Intern. Symp. on Turbo Codes and Related Topics*, (Brest, France), pp. 75–82, Sept. 1–5 2003.

[10] P. O. Vontobel and R. Koetter, "Graph-cover decoding and finite-length analysis of message-passing iterative decoding of LDPC codes," *submitted to IEEE Trans. Inform. Theory, available online under* `http://www.arxiv.org/abs/cs.IT/0512078`, Dec. 2005.

[11] J. Feldman, T. Malkin, C. Stein, R. A. Servedio, and M. J. Wainwright, "LP decoding corrects a constant fraction of errors," in *Proc. IEEE Intern. Symp. on Inform. Theory*, (Chicago, IL, USA), p. 68, June 27–July 2 2004.

[12] R. G. Gallager, *Low-Density Parity-Check Codes*. M.I.T. Press, Cambridge, MA, 1963. Available online under `http://web.mit.edu/gallager/www/pages/ldpc.pdf`.