Linear Programming Decoding of Spatially Coupled Codes

Louay Bazzi, Badih Ghazi, and Rüdiger L. Urbanke

Abstract—**For a given family of spatially coupled codes, we prove that the linear programming (LP) threshold on the binary-symmetric channel (BSC) of the tail-biting graph cover ensemble is the same as the LP threshold on the BSC of the derived spatially coupled ensemble. This result is in contrast with the fact that spatial coupling significantly increases the belief propagation threshold. To prove this, we establish some properties related to the dual witness for LP decoding. More precisely, we prove that the existence of a dual witness, which was previously known to be sufficient for LP decoding success, is also necessary and is equivalent to the existence of certain acyclic hyperflows. We also derive a sublinear (in the block length) upper bound on the weight of any edge in such hyperflows, both for regular low-density parity-check (LPDC) codes and spatially coupled codes and we prove that the bound is asymptotically tight for regular LDPC codes. Moreover, we show how to trade crossover probability for LP excess on all the variable nodes, for any binary linear code.**

Index Terms—**Linear programming (LP) decoding, spatially-coupled codes, binary-symmetric channel (BSC), low-density parity-check (LDPC) codes, factor graphs.**

I. Introduction

In recent years, Linear Programming (LP) decoding has been extensively studied as a potential approach to decoding an arbitrary binary linear code when transmitting over a noisy communication channel. Following its introduction in [11], it has been shown to have a good performance in different setups. For instance, LP decoding corrects a constant fraction of errors on certain codes [9] and achieves capacity on a wide range of probabilistic channels [10]. In parallel, spatial coupling emerged as a successful method for designing capacity-achieving channel-coding schemes since its introduction by [14]. In particular, recent work by [17] and [18] showed that spatially coupled codes significantly improve the performance of BP decoding on any binary-input memoryless output-symmetric channel.

Manuscript received March 4, 2013; revised January 27, 2014; accepted May 1, 2014. Date of publication June 2, 2014, date of current version July 10, 2014. This paper was presented at the 2013 International Symposium on Information Theory [BGU13].

L. Bazzi is with the Department of Electrical and Computer Engineering, American University of Beirut, Beirut 1107 2020, Lebanon (e-mail: lb13@aub.edu.lb).

B. Ghazi was with the American University of Beirut, Beirut 1107 2020, Lebanon. He is now with the Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, Cambridge, MA 02139 USA (e-mail: badih@mit.edu).

R. Urbanke is with the Department of Electrical and Computer Engineering, École Polytechnique Fédérale de Lausanne, Lausanne 1015, Switzerland (e-mail: ruediger.urbanke@epfl.ch).

Communicated by D. Burshtein, Associate Editor for Coding Techniques.

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TIT.2014.2325903

However, the performance of LP decoding on spatially coupled codes has remained elusive. In this work, we initiate this study by proving that for a given family of spatially coupled codes, the LP threshold on the BSC of the tail-biting graph cover ensemble is the same as the LP threshold on the BSC of the derived spatially coupled ensemble. (Roughly speaking, a tail-biting graph cover code is a "circular version" of a spatially coupled code. See Section II for the formal definition of the tail-biting graph cover ensemble and for some illustrating figures.) This result is in contrast with the fact that spatial coupling significantly increases the Belief Propagation (BP) threshold as shown in [17] and [18].

In the remainder of this introductory section, we give some background on binary linear codes, LP decoding and spatially coupled codes. We then state our technical contributions and outline the remaining parts of the paper.

A. Binary Linear Codes

A binary linear code ζ of block length n is a subspace of the $\mathbb{F}_2$-vector space $\mathbb{F}_2^n$. The ϵ-BSC (Binary Symmetric Channel) with input $X \in \mathbb{F}_2^n$ and output $Y \in \mathbb{F}_2^n$ flips each input bit independently with probability ϵ. Let γ be the log-likelihood ratio vector which is given by $\gamma_i = \log\left(\frac{p_{Y_i|X_i}(y_i|0)}{p_{Y_i|X_i}(y_i|1)}\right) = (-1)^{y_i}\log\frac{1-\epsilon}{\epsilon}$ for any $i \in \{1,\dots,n\}$. The optimal decoder is the Maximum Likelihood (ML) decoder which is given by

$$
\begin{aligned}
\hat{x}_{\mathrm{ML}} &= \underset{x\in\zeta}{\operatorname{argmax}}\ p_{Y|X}(y|x) \\
&= \underset{x\in\zeta}{\operatorname{argmax}} \prod_{i=1}^{n} p_{Y_i|X_i}(y_i|x_i) \\
&= \underset{x\in\zeta}{\operatorname{argmax}} \frac{\prod_{i=1}^{n} p_{Y_i|X_i}(y_i|x_i)}{\prod_{i=1}^{n} p_{Y_i|X_i}(y_i|0)} \\
&= \underset{x\in\zeta}{\operatorname{argmax}} \log\left(\prod_{i=1}^{n}\frac{p_{Y_i|X_i}(y_i|x_i)}{p_{Y_i|X_i}(y_i|0)}\right) \\
&= \underset{x\in\zeta}{\operatorname{argmax}} \sum_{i=1}^{n}\log\left(\frac{p_{Y_i|X_i}(y_i|x_i)}{p_{Y_i|X_i}(y_i|0)}\right) \\
&= \underset{x\in\zeta}{\operatorname{argmin}} \sum_{i=1}^{n}\gamma_i x_i
\end{aligned}
$$

where the second equality follows from the fact that the channel is memoryless. Since the objective function is linear in x, replacing ζ by the convex hull $\text{conv}(\zeta)$ of ζ does not change the value of the minimal solution. Hence, we get

$$\hat{x}_{\text{ML}} = \underset{x \in \text{conv}(\zeta)}{\text{argmin}} \sum_{i=1}^{n} \gamma_i x_i \tag{1}$$

ML decoding is known to be NP-hard for general binary linear codes [3]. This motivates the study of suboptimal decoding algorithms that have small running times.

B. Linear Programming Decoding

LP decoding was introduced by [11], [8] and is based on the idea of replacing $\text{conv}(\zeta)$ in (1) with a larger subset of $\mathbb{R}^n$, with the goal of reducing the running time while maintaining a good error correction performance. First, note that $\text{conv}(\zeta) = \text{conv}(\bigcap_{j \in C} \zeta_j)$ where $\zeta_j = \{z \in \{0,1\}^n : wt(z|_{N(j)}) \text{ is even}\}$[1] for all j in the set C of check nodes corresponding to a fixed Tanner graph of ζ and where $N(j)$ is the set of all neighbors of check node j. Then, LP decoding is given by relaxing $\text{conv}(\bigcap_{j \in C} \zeta_j)$ to $\bigcap_{j \in C} \text{conv}(\zeta_j)$:

$$\hat{x}_{LP} = \underset{x \in P}{\text{argmin}} \sum_{i=1}^{n} \gamma_i x_i \tag{2}$$

where $P = \bigcap_{j \in C} \text{conv}(\zeta_j)$ is the so-called "fundamental polytope" that will be carefully considered in the proof of Theorem 3.2. A central property of P is that, in the case of LDPC codes, it can be described by a number of inequalities that is linear in n, which implies that the linear program (2) can be solved in time polynomial in n using the ellipsoid algorithm or interior point methods.

When analyzing the operation of LP decoding, one can assume that the all-zeros codeword was transmitted [11]. Then, by normalizing the expression for the log-likelihood ratio γ given in Section I-A by the positive constant $\log(\frac{1-\epsilon}{\epsilon})$, we can assume that the log-likelihood ratio is given by $\gamma_i = 1$ if $y_i = 0$ and $\gamma_i = -1$ if $y_i = 1$ for all $i \in \{1, \ldots, n\}$. As in previous work, we make the conservative assumption that LP decoding fails whenever there are multiple optimal solutions to the linear program (2). In other words, under the all-zeros codeword assumption, LP decoding succeeds if and only if the zero codeword is the unique optimal solution to the linear program (2). In order to show that LP decoding corrects a constant fraction of errors when the Tanner graph has sufficient expansion, [9] introduced the concept of a dual witness, which is a dual feasible solution with zero cost and with a given set of constraints having a positive slack. By complementary slackness, it follows that the existence of a dual witness implies LP decoding success [9]. A simplified (but equivalent) version of this dual witness, called a hyperflow, was introduced in [6] (and later generalized in [13]) and used to prove that LP decoding can correct a larger fraction of errors in a probabilistic setting. This hyperflow will be described in Section III. However, it was unkown whether the existence of a hyperflow (or equivalently that of a dual witness) is necessary for LP decoding success. We will show, by careful consideration of the fundamental polytope P, that this is indeed the case.

[1] For $x \in \{0,1\}^n$ and $S \subseteq \{1, \ldots, n\}$, $x|_S \in \{0,1\}^n$ denotes the restriction of x to S, i.e., $(x|_S)_i = x_i$ if $i \in S$ and $(x|_S)_i = 0$ otherwise, and $wt(x)$ denotes the Hamming weight of x.

C. Spatially Coupled Codes

The idea of spatial coupling has been recently used in coding theory, compressive sensing and other fields. Spatially coupled codes (or convolutional LDPC codes) were introduced in [14]. Recently, [17] showed that the BP threshold of spatially coupled codes is the same as the MAP (Maximum A posteriori Probability) threshold of the base LDPC code in the case of the Binary Erasure Channel (BEC). Moreover, [18] showed that spatially coupled codes achieve capacity under belief propagation. In compressive sensing, [7], [15], [16] showed that spatial coupling can be used to design dense sensing matrices that achieve, under an approximate message passing algorithm, the same peformance as the optimal l_0-norm minimizing compressive sensing decoder. In coding theory, the intuition behind the improvement in performance due to spatial coupling is that the check nodes located at the boundaries have low degrees which enables the BP algorithm to initially recover the transmitted bits at the boundaries. Then, the other transmitted bits are progressively recovered from the boundaries to the center of the code. A similar intuition is behind the good performance of spatial coupling in compressive sensing [7].

D. Contributions

We prove that the LP threshold of the spatially coupled ensemble on the BSC is the same as that of the tail-biting graph cover ensemble (Theorem 9.1). To do so, we prove some general results about LP decoding of LDPC codes that may be of independent interest:

1) We prove that the existence of a dual witness which was previously known to be sufficient for LP decoding success is also necessary and is equivalent to the existence of certain acyclic hyperflows (Theorem 3.2).
2) We derive a sublinear (in the block length) upper bound on the weight of any edge in the hyperflow, for regular LDPC codes (Theorem 5.1) and spatially coupled codes (Theorem 6.1). In the regular case, we show that our bound is asymptotically tight (Theorem 5.11).
3) We show how to trade crossover probability for "LP excess" on all the variable nodes, for any binary linear code (Theorem 8.1).

E. Outline

The paper is organized as follows. In Section II, we formally state the main result of the paper and give a high-level sketch of the proof. In Section III, we prove that the existence of a dual witness which was previously known to be sufficient for LP decoding success is also necessary and is equivalent to the existence of certain weighted directed acyclic graphs. In Section IV, we show how to transform those weighted directed acyclic graphs into weighted directed forests while preserving their central properties. In Section V, we prove, using the result of Section IV, a sublinear (in the block length) upper bound on the weight of any edge in such graphs, for regular codes. An analogous upper bound is proved in Section VI for spatially coupled codes. In Section VII, we relate LP decoding on a tail-biting graph cover code and on a spatially coupled code. In Section VIII, we show how to trade crossover probability for "LP excess" on all the variable nodes, for any binary linear code. The results of Sections VI, VII and VIII are finally used in Section IX, where we prove the main result of the paper.

F. Notation and Terminology

We denote the set of all non-negative integers by $\mathbb{N}$. For any integers n, a, b with $n \geq 1$, we denote by $[n]$ the set $\{1, \ldots, n\}$ and by $[a : b]$ the set $\{a, \ldots, b\}$. For any event A, let $\overline{A}$ be the complement of A. For any vertex v of a graph G, we let $N(v)$ denote the set of all neighbors of v in G. For any $x \in \{0, 1\}^n$ and any $S \subseteq [n]$, let $x|_S \in \{0, 1\}^n$ s.t. $(x|_S)_i = x_i$ if $i \in S$ and $(x|_S)_i = 0$ otherwise. A binary linear code ζ can be fully described as the nullspace of a matrix $H \in \mathbb{F}_2^{r \times n}$ (with $r \geq n-k$), called the parity check matrix of ζ. For a fixed H, ζ can be graphically represented by a Tanner graph (V, C, E) which is a bipartite graph where $V = \{v_1, \ldots, v_n\}$ is the set of variable nodes, $C = \{c_1, \ldots, c_r\}$ is the set of check nodes and for any $i \in [n]$ and any $j \in [r]$, $(v_i, c_j) \in E$ if and only if $H_{j,i} = 1$. If H is sparse, then ζ is called a Low-Density Parity-Check (LDPC) code. LDPC codes were introduced and first analyzed by Gallager [12]. If the number of ones in each column of H is d_{v} and the number of ones in each row of H is d_{c}, ζ is called a $(d_{\mathrm{v}}, d_{\mathrm{c}})$-regular code. We let $\hat{d}_{\mathrm{v}} = (d_{\mathrm{v}} - 1)/2$. Throughout the paper, we assume that $n, d_{\mathrm{c}}, d_{\mathrm{v}} > 2$.

II. Main Result

First, we define the spatially coupled codes under consideration.

Definition 2.1 (Spatially Coupled Code): A $(d_{\mathrm{v}}, d_{\mathrm{c}} = kd_{\mathrm{v}}, L, M)$ spatially coupled code, with d_{v} an odd integer and M divisible by k, is constructed by considering the index set $[-L - \hat{d}_{\mathrm{v}} : L + \hat{d}_{\mathrm{v}}]$ and satisfying the following conditions:[2]

1) *M variable nodes are placed at each position in $[-L : L]$ and $M\frac{d_{\mathrm{v}}}{d_{\mathrm{c}}}$ check nodes are placed at each position in $[-L - \hat{d}_{\mathrm{v}} : L + \hat{d}_{\mathrm{v}}]$.*
2) *For any $j \in [-L+\hat{d}_{\mathrm{v}} : L-\hat{d}_{\mathrm{v}}]$, a check node at position j is connected to k variable nodes at position $j+i$ for all $i \in [-\hat{d}_{\mathrm{v}} : \hat{d}_{\mathrm{v}}]$.*
3) *For any $j \in [-L - \hat{d}_{\mathrm{v}} : -L + \hat{d}_{\mathrm{v}} - 1]$, a check node at position j is connected to k variable nodes at position i for all $i \in [-L : j + \hat{d}_{\mathrm{v}}]$.*
4) *For any $j \in [L - \hat{d}_{\mathrm{v}} + 1 : L + \hat{d}_{\mathrm{v}}]$, a check node at position j is connected to k variable nodes at position i for all $i \in [j - \hat{d}_{\mathrm{v}} : L]$.*
5) *No two check nodes at the same position are connected to the same variable node.*[3]

[2]Informally, $2L + 1$ is the number of "layers" and M is the number of variable nodes per "layer".

[3]This implies that for any $i \in [-L : L]$, a variable node at position i is connected to exactly one check node at position $i+j$ for every $j \in [-\hat{d}_{\mathrm{v}} : \hat{d}_{\mathrm{v}}]$. This also implies that every variable node has degree d_{v}.

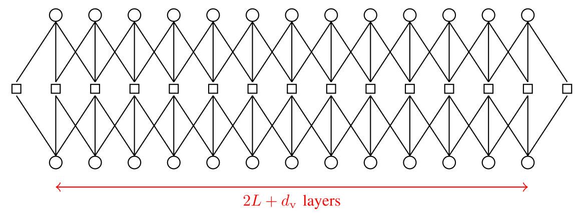

Fig. 1. Example of a spatially coupled codes with $2L + d_{\mathrm{v}}$ vertical layers.

An example of a spatially-coupled code is given in Figure 1. With the exception of the non-degeneracy condition 5, Definition 2.1 above is the same as that given in Section II-A of [17]. We next define the tail-biting graph cover codes under consideration which are similar to the tail-biting LDPC convolutional codes introduced by [22].

Definition 2.2 (Tail-Biting Graph Cover Code): A $(d_{\mathrm{v}}, d_{\mathrm{c}} = kd_{\mathrm{v}}, L, M)$ tail-biting graph cover code, with d_{v} an odd integer and M divisible by k, is constructed by considering the index set $[-L : L]$ and satisfying the following conditions:

1) *M variable nodes and $M\frac{d_{\mathrm{v}}}{d_{\mathrm{c}}}$ check nodes are placed at each position in $[-L : L]$.*
2) *For any $j \in [-L : L]$, a check node at position j is connected to k variable nodes at position $(j + i) \mod [-L : L]$ for all $i \in [-\hat{d}_{\mathrm{v}} : \hat{d}_{\mathrm{v}}]$.*
3) *No two check nodes at the same position are connected to the same variable node.*[4]

Figure 2 shows the construction of a tail-biting graph cover code with L layers. Note that "cutting" a tail-biting graph cover code at any position $i \in [-L : L]$ yields a spatially coupled code. This motivates the following definition.

Definition 2.3 (Derived Spatially Coupled Codes): Let ζ be a $(d_{\mathrm{v}}, d_{\mathrm{c}} = kd_{\mathrm{v}}, L, M)$ tail-biting graph cover code. For each $i \in [-L : L]$, the $(d_{\mathrm{v}}, d_{\mathrm{c}} = kd_{\mathrm{v}}, L - \hat{d}_{\mathrm{v}}, M)$ spatially coupled code ζ'_i is obtained from ζ by removing all M variable nodes and their adjacent edges at each position $i + j \mod [-L : L]$ for every $j \in [0 : 2\hat{d}_{\mathrm{v}} - 1]$. Then, $\mathcal{D}(\zeta) = \{\zeta'_{-L}, \ldots, \zeta'_L\}$ is the set of all $2L + 1$ derived spatially coupled codes of ζ.

Figure 3 shows how to obtain a derived spatially coupled code.

Definition 2.4 (Ensembles and Thresholds): Let Γ be an ensemble, i.e., a probability distribution over codes. The LP threshold ξ of Γ on the BSC is defined as $\xi = \sup\{\epsilon > 0 \mid \Pr_{\substack{\zeta\sim\Gamma \\ \epsilon\text{-}BSC}}[\text{LP decoding error on } \zeta] = o(1)\}$.[5]

We are now ready to state the main result of this paper.

Theorem 2.5 (Main Result: $\xi_{GC} = \xi_{SC}$): Let Γ_{GC} be a $(d_{\mathrm{v}}, d_{\mathrm{c}} = kd_{\mathrm{v}}, L, M)$ tail-biting graph cover ensemble with d_{v} an odd integer and M divisible by k. Let Γ_{SC} be the $(d_{\mathrm{v}}, d_{\mathrm{c}} = kd_{\mathrm{v}}, L-\hat{d}_{\mathrm{v}}, M)$ spatially coupled ensemble which is sampled by choosing a tail-biting graph cover code $\zeta \sim \Gamma_{GC}$

[4]This implies that for all $i \in [-L : L]$, a variable node at position i is connected to exactly one check node at position $(i + j) \mod [-L : L]$ for each $j \in [-\hat{d}_{\mathrm{v}} : \hat{d}_{\mathrm{v}}]$. This also implies that every variable node has degree d_{v}.

[5]Here, the $o(1)$ is w.r.t. the block length n of the codes in the ensemble Γ. In the case of spatially coupled codes and tail-biting graph cover codes where d_{v} and d_{c} are constants, we have that $n = O(L \times M)$ where we will express M as a function of L. Thus, in our case, the $o(1)$ can be equivalently be taken to be w.r.t. L.

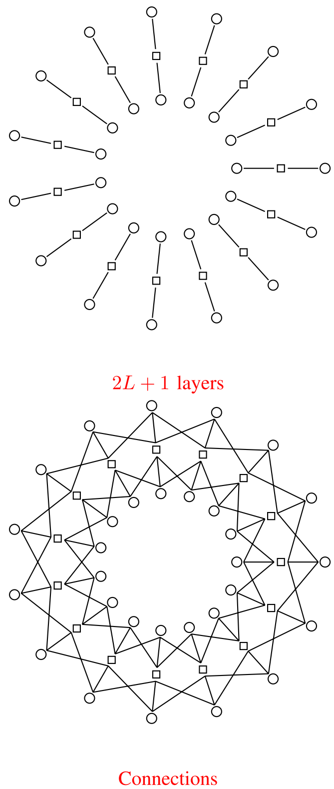

Fig. 2. Construction of a tail-biting graph cover ensemble.

and returning a element of $\mathcal{D}(\zeta)$ chosen uniformly at random[6]*. Denote by ξ_{GC} and ξ_{SC} the respective LP threholds of Γ_{GC} and Γ_{SC} on the BSC. Then, there exists $\nu > 0$ depending only on d_{v} and d_{c} s.t. if $M = o(L^{\nu})$ and Γ_{SC} satisfies the property that for any constant $\Delta > 0$,*

$$\Pr_{\substack{\zeta' \sim \Gamma_{SC} \\ (\xi_{SC}-\Delta)\text{-}BSC}} [\textit{LP decoding error on } \zeta'] = o(\frac{1}{L^2}) \quad (3)$$

then, $\xi_{GC} = \xi_{SC}$.

Note that for $M = \omega(\log L)$, condition (3) above is expected to hold for the spatially coupled ensemble Γ_{SC} since under iterative decoding algorithms, the error probability on the $(\xi_{SC} - \Delta)$-BSC is expected to decay to zero as $O(Le^{-c \times \Delta^2 \times M})$ for some constant $c > 0$. Moreover, note that in the regime $M = \Theta(L^{\delta})$ (for any positive constant δ), spatial coupling provides empirical improvements under iterative decoding and in fact, the improvement is expected to take place as long as L is subexponential in M [20].

A. High Level Sketch of the Proof

The main part of the proof is to show that $\xi_{GC} \geq \xi_{SC}$. We need to show that, for any $\epsilon \leq \xi_{SC}$, the LP decoder succeeds with high probability on the tail-biting graph cover code when we transmit on the ϵ-BSC. Since $\epsilon \leq \xi_{SC}$, when transmitting on a random spatially-coupled code over the ϵ-BSC, the LP decoder succeeds with high probability. First, we show that this LP decoding success implies the existence of a dual witness (Theorem 3.2). Then, we prove that the maximum weight of an edge in an acyclic version of this dual witness is sublinear in the block length (Theorem 6.1). We next show that if we instead transmit on a derived spatially-coupled code over the $(\epsilon - \delta)$-BSC, then with high probability, there exists a dual witness with slack at least $\delta/2$ in all the variable node inequalities (Theorem 8.1). We finally use this slack along with the sublinear upper bound on the edge weight in order to prove that the average of the dual witnesses for each of the $2L+1$ derived spatially coupled codes forms a dual witness for the tail-biting graph cover code (Proof of Theorem 9.1). Thus, we conclude that, with high probability, there is LP decoding success on the tail-biting graph cover code.

On a high level, the reason why LP decoding does not benefit from spatial coupling is the sublinear (in the block length) upper bound on the edge weight in an acyclic dual witness. This sublinear upper bound prevents the correction that might be taking place at the boundaries of the code from significantly propagating toward the center of the code. Such a propagation was at the basis of the improvement in the performance of BP due to spatial coupling.

[6]Here, $\mathcal{D}(\zeta)$ refers to Definition 2.3.

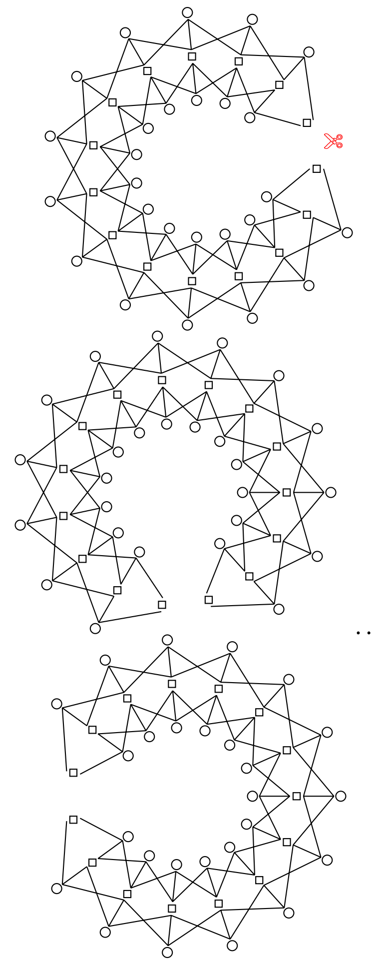

Fig. 3. Derived spatially coupled codes: This figure shows 3 out of the $2L+1$ spatially coupled codes that are derived from the tail-biting graph cover code constructed in Figure 2.

III. LP Decoding, Dual Witnesses, Hyperflows and WDAGs

The dual of the LP decoder was first examined in [9] (and further studied in [4], [23] and [24]). The following definition is based on Definition 1 of [9].

Definition 3.1 (Dual Witness): For a given Tanner graph $\mathcal{T} = (V, C, E)$ and a (possibly scaled) log-likelihood ratio function $\gamma : V \to \mathbb{R}$, a dual witness w is a function $w : E \to \mathbb{R}$ that satisfies the following 2 properties:

$$\forall v \in V : \sum_{\substack{c \in N(v): \\ w(v,c)>0}} w(v,c) < \sum_{\substack{c \in N(v): \\ w(v,c)\leq 0}} (-w(v,c)) + \gamma(v) \quad (4)$$

$$\forall c \in C, \forall v, v' \in N(c) : \ w(v,c) + w(v',c) \geq 0 \quad (5)$$

The following theorem relates the existence of a dual witness to LP decoding success. The fact that the existence of a dual witness implies LP decoding success was shown in [9]. We prove that the converse of this statement is also true. This converse will be used in the proof of Theorem 8.1.

Theorem 3.2 (Existence of a Dual Witness and LP Decoding Success): Let $\mathcal{T} = (V, C, E)$ be a Tanner graph of a binary linear code with block length n and let $\eta \in \{0,1\}^n$ be any error pattern. Then, there is LP decoding success for η on $\mathcal{T}$ if and only if there is a dual witness for η on $\mathcal{T}$.

Proof of Theorem 3.2: See Appendix -A. □

Remark 3.3: The proof of Theorem 3.2 holds for non-binary error patterns, i.e., for arbitrary log-likelihood ratios.

The following definition is based on Definition 1 of [6].

Definition 3.4 (Hyperflow): For a given Tanner graph $\mathcal{T} = (V, C, E)$ and a (possibly scaled) log-likelihood ratio function $\gamma : V \to \mathbb{R}$, a hyperflow w is a function $w : E \to \mathbb{R}$ that satisfies property (4) above as well as the following property:

$$\forall c \in C, \exists P_c \geq 0, \exists v \in N(c) \text{ s.t. } w(v,c) = -P_c$$
$$\text{and } \forall v' \in N(c) \text{ s.t. } v' \neq v, w(v',c) = P_c \quad (6)$$

By Proposition 1 of [6], the existence of a hyperflow is equivalent to that of a dual witness. Hence, by Theorem 3.2 above, we get:

Corollary 3.5 (Existence of a Hyperflow and LP Decoding Success): Let $\mathcal{T} = (V, C, E)$ be a Tanner graph of a binary linear code with block length n and let $\eta \in \{0,1\}^n$ be any error pattern. Then, there is LP decoding success for η on $\mathcal{T}$ if and only if there is a hyperflow for η on $\mathcal{T}$.

Definition 3.6 (WDG and WDAG Corresponding to a Hyperflow or a Dual Witness): Let $\mathcal{T} = (V, C, E)$ be a Tanner graph, $\gamma : V \to \mathbb{R}$ a (possibly scaled) log-likelihood ratio function and $w : E \to \mathbb{R}$ a dual witness or a hyperflow. The weighted directed graph (WDG) (V, C, E, w, γ) associated with $\mathcal{T}$,γ and w has vertex set $V \cup C$ and for any $v \in V$ and any $c \in C$, an arrow is directed from v to c if $w(v,c) > 0$, an arrow is directed from c to v if $w(v,c) < 0$ and v and c are not connected by an arrow if $w(v,c) = 0$. Moreover, a directed edge between $v \in V$ and $c \in C$ has weight $|w(v,c)|$. If the arrows of (V, C, E, w, γ) contain no directed cycles, then (V, C, E, w, γ) is said to be weighted directed acyclic graph (WDAG).

The following theorem shows that whenever there exists a WDG corresponding to a hyperflow or a dual witness, there also exists a WDAG corresponding to a hyperflow.

Algorithm 1 Transforming the Dual Witness WDG G for γ Into a Hyperflow WDAG G'' for γ

Input: $G = (V, C, E, w, \gamma)$
Output: $G'' = (V, C, E, w'', \gamma)$

```
G' = (V, C, E, w', γ) ← G
while G' has a directed cycle do
    c ← any directed cycle of G'
    w_min ← minimum weight of an edge of c   ▷ All edges
        along c have a positive weight.
    Subtract w_min from the weights of all edges of c
    Remove all zero weight edges
    Store the resulting WDG in G'
end while

for all j ∈ C do
    d(j) ← degree of j
    {v_1, ..., v_d(j)} ← neighbours of j in order of increasing
        w'(v_i, j)
    if w'(v_1, j) ≥ 0 then   ▷ All edges are directed toward j
        and can thus be removed.
        w''(v_i, j) ← 0 ∀i ∈ [d(j)]
    else   ▷ (v_1, j) is the only edge directed away from j.
        w''(v_1, j) ← w'(v_1, j)
        w''(v_i, j) ← |w'(v_1, j)| ∀i ∈ {2, ..., d(j)}
    end if
end for
```

Theorem 3.7 (Existence of a WDAG): Let $\mathcal{T} = (V, C, E)$ be a Tanner graph of a binary linear code with block length n and let $\eta \in \{0,1\}^n$ be any error pattern. If $G = (V, C, E, w, \gamma)$ is a WDG (Weighted Directed Graph) corresponding to a dual witness for η on $\mathcal{T}$, then there is an WDAG $G'' = (V, C, E, w'', \gamma)$ corresponding to a hyperflow for η on $\mathcal{T}$.

Before proving Theorem 3.7, we summarize the different characterizations of LP decoding success.

Theorem 3.8: Let $\mathcal{T} = (V, C, E)$ be a Tanner graph of a binary linear code with block length n and let $\eta \in \{0,1\}^n$ be any error pattern. Then, the following are equivalent:

1) *There is LP decoding success for η on $\mathcal{T}$.*
2) *There is a dual witness for η on $\mathcal{T}$.*
3) *There is a hyperflow for η on $\mathcal{T}$.*
4) *There is a WDAG for η on $\mathcal{T}$.*

In order to prove Theorem 3.7, we give an algorithm that transforms a WDG G satisfying Equations (4) and (5) into an acyclic WDG G'' satisfying Equations (4) and (6). The description of this algorithm is given in Algorithm 1.

The output of Algorithm 1 on a particular input is given in Figure 4.

The next lemma is used to complete the proof of Theorem 3.7.

Lemma 3.9: After each iteration of the while loop of Algorithm 1, we have

(I) *The number of cycles of G' decreases by at least* 1.
(II) *G' satsifies the dual witness equations (4) and (5).*

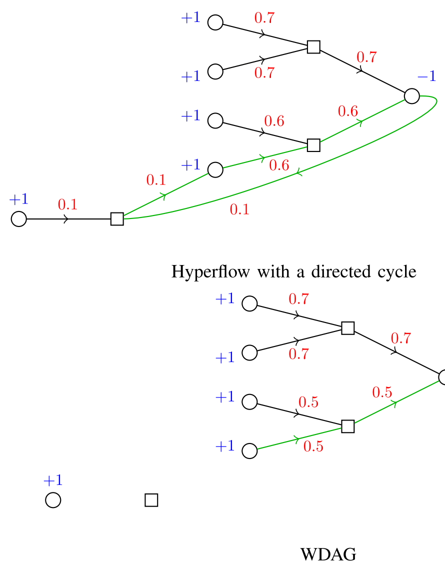

Fig. 4. Output of Algorithm 1 on a given input.

Proof of Lemma 3.9: (I) follows from the fact that cycle c is being broken in every iteration of the while loop and no new cycle is added by reducing the absolute weights of some edges of the WDG. (II) follows from the fact that during any iteration of the while loop, we are possibly repeatedly reducing the absolute weights of one ingoing and one outgoing edge of a variable or check node by the same amount, which maintains the original LP constraints (4) and (5). □

Proof of Theorem 3.7: First, note that the while loop of Algorithm 1 will be executed a number of times no larger than the number of cycles of G, which is finite. By Lemma 3.9, after the last iteration of the while loop, G' is an acyclic WDG that satisfies (4) and (5). The for loop of Algorithm 1 decreases the weights of edges that are directed away from variable nodes; thus, it maintains (4) and G'' inherits the acyclic property of G'. Moreover, G'' satsifies (6), which completes the proof Theorem 3.7. □

Remark 3.10: We note the following:

- *If we let G be a WDG of a $(d_{\mathrm{v}}, d_{\mathrm{c}})$-regular code and let G' be the WDAG obtained by running Algorithm 1 on G, then each check node of G' has either degree d_{c} or degree 0, and every variable node of G' has degree at most d_{v}.*
- *In virtue of Theorem 3.2, Theorem 3.5 and Theorem 3.7, we will use the terms "hyperflow", "dual witness" and "WDAG" interchangeably in the rest of this paper.*

IV. Transforming a WDAG Into a Directed Weighted Forest

The WDAG corresponding to a hyperflow has no directed cycles but it possibly has cycles when viewed as an undirected graph. In this section, we show how to transform the WDAG corresponding to a hyperflow into a directed weighted forest (which is by definition a directed graph that is acyclic even when viewed as an undirected graph). This forest has possibly a larger number of variable and check nodes than the original WDAG but it still satisfies Equations (4) and (6). Each variable node v' of the forest will correspond to one variable node v of the original WDAG. Similarly, each check node c' of the forest will correspond to one check node c of the original WDAG. Moreover, the set of variable nodes of the forest corresponding to a particular variable node v of the original WDAG will have their weights sum up to the original weight of v.[7] Furthermore, the directed paths of the forest will be in a bijective correspondence with the directed paths of the original WDAG. This transformation will be used when we derive an upper bound on the weight of an edge in a WDAG of a $(d_{\mathrm{v}}, d_{\mathrm{c}})$-regular LDPC code in Section V and of a spatially coupled code in Section VI.

Theorem 4.1 (Transforming a WDAG Into a Directed Weighted Forest): Let $G = (V, C, E, w, \gamma)$ be a WDAG. Then, G can be transformed into a directed weighted forest $T = (V', C', E', w', \gamma')$ that has the following properties:

1) *$V' = \bigcup_{v \in V} V'_v$ where $V'_x \cap V'_y = \emptyset$ for all $x, y \in V$ s.t. $x \neq y$. For every $v \in V$, each variable node in V'_v is called a "replicate" of v.*
2) *$C' = \bigcup_{c \in C} C'_c$ where $C'_x \cap C'_y = \emptyset$ for all $x, y \in C$ s.t. $x \neq y$. For every $c \in C$, each check node in C'_c is called a "replicate" of c.*
3) *For all $v \in V$, $\sum_{v' \in V'_v} \gamma'(v') = \gamma(v)$.*
4) *For all $v \in V$ and all $v' \in V_v$, $\gamma'(v')$ has the same sign as $\gamma(v)$.*
5) *The forest T satisfies the hyperflow equations (4) and (6).*
6) *The directed paths of G are in a bijective correspondence with the directed paths of T. Moreover, if the directed path h' of T corresponds to the directed path h of G, then the variable and check nodes of h' are replicates of the corresponding variable and check nodes of h. For instance, if*
$$h = v_1 \to c_1 \to \cdots \to v_l \to c_l \to v_{l+1}$$
and
$$h' = v'_1 \to c'_1 \to \cdots \to v'_l \to c'_l \to v'_{l+1}$$
then $v'_i \in V'_{v_i}$ for all $i \in [l+1]$ and $c'_i \in C'_{c_i}$ for all $i \in [l]$ where V'_{v_i} and C'_{c_i} are given in 1. and in 2. above respectively.
7) *If G has a single sink node with a single incoming edge that has weight α, then T has a single sink node with a single incoming edge and that has the same weight α.*

The proof of Theorem 4.1 (given in Appendix -B) is based on the following algorithm which transforms the WDAG G into the directed weighted forest T.

Note that the notion of "ancestors" in Algorithm 2 is with respect to the directions of the arrows of the WDAG. A sample execution of this algorithm is shown in Figure 5 and Figure 6.

The analysis of Algorithm 2 and the proof of Theorem 4.1 are given in Appendix -B.

[7]In a WDAG $G = (V, C, E, w, \gamma)$, the weight of vertex $v \in V$ is $\gamma(v)$.

[8]A topological ordering of a directed graph is an ordering of its vertices s.t. for every directed edge (u, v), u comes before v in the ordering.

Algorithm 2 Transforming the WDAG G Into the Directed Weighted Forest T

Input: $G = (V, C, E, w, \gamma)$
Output: $T = (V', C', E', w', \gamma')$

for each $v \in V$ taken in topological order[8] **do**
  $p \leftarrow$ number of outgoing edges of v
  $\{e_j^{(v)}\}_{j=1}^{p} \leftarrow$ weights of outgoing edges of v
  $e_T^{(v)} \leftarrow \sum_{j=1}^{p} e_j^{(v)}$
  Create p replicates of the subtree rooted at v ▷ Contains all ancestors of v in the current WDAG
  for each $l \in [p]$ **do**
    Scale the lth subtree by $e_l^{(v)}/e_T^{(v)}$ ▷ The weights of all variable nodes and edges are scaled
    Connect the lth subtree to the lth outgoing edge of v
  end for
end for

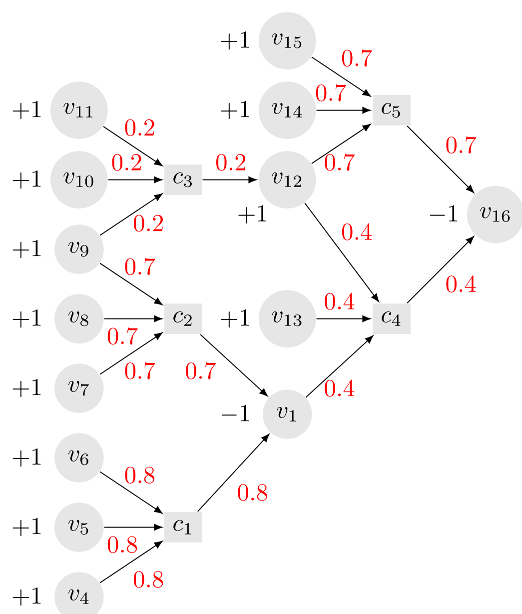

Fig. 5. Input to Algorithm 2 which is a $(d_v = 3, d_c = 4)$-regular graph. The labels on the edges correspond to the $e_j^{(v)}$ variables in the description of Algorithm 2. See Figure 6 for the output of Algorithm 2 on this input.

V. Maximum Weight of an Edge in a WDAG of a Regular Code on the BSC

In this section, we present a sublinear (in the block length n) upper bound on the weight of an edge in a WDAG of a regular code. On a high level, the reason why such an upper bound will be useful to us in later sections is that the larger the edge weight can be, the easier it is for variables received correctly to help correct variables in error that are located far away in the graph. The main idea of the proof of the sublinear upper bound on the edge weight in a WDAG of a regular code is the following. Consider a WDAG G of a (d_v, d_c)-regular LDPC code. Note that each variable node has a log-likelihood ratio of ± 1. Thus, the total amount of flow available in the

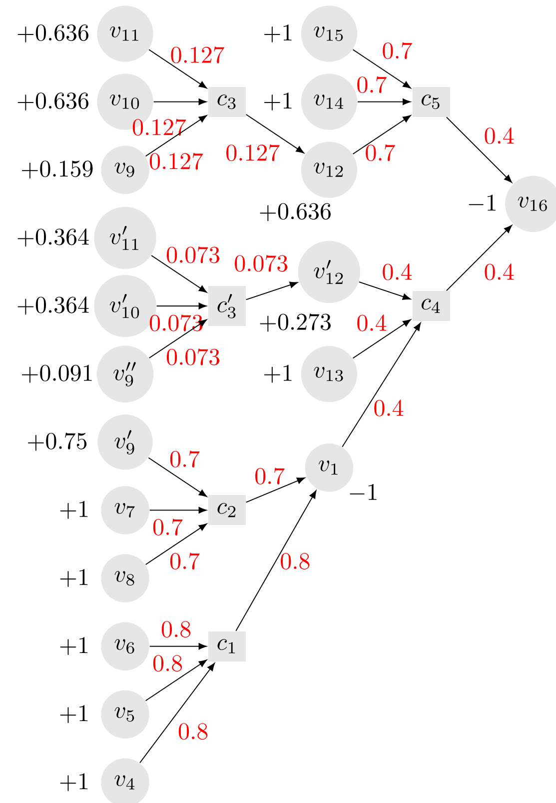

Fig. 6. Output of Algorithm 2 on the input given in Figure 5. Note that v_9, v'_9 and v''_9 are replicates of each other and that v''_9 is created in the iteration corresponding to node v_{12}.

WDAG is most n. Moreover, for a substantial weight to get "concentrated" on an edge in the WDAG, the +1's should "move" from variable nodes across the WDAG toward that edge. By the hyperflow equation (6), each check node cuts its incoming flow by a factor of $d_c - 1$. Thus, it can be seen that the maximum weight that can get concentrated on an edge is asymptotically smaller than n.

Theorem 5.1 (Maximum Weight of an Edge in the WDAG of a Regular Code on the BSC): Let $G = (V, C, E, w, \gamma)$ be a WDAG corresponding to LP decoding of a (d_v, d_c)-regular LDPC code (with $d_v, d_c > 2$) on the BSC. Let $n = |V|$ and $\alpha_{max} = \max_{e \in E} |w(e)|$ be the maximum weight of an edge in G. Then,

$$\alpha_{max} \leq c n^{\frac{\ln(d_v - 1)}{\ln(d_v - 1) + \ln(d_c - 1)}} = o(n) \tag{7}$$

for some constant $c > 0$ depending only on d_v.

We now state and prove a series of lemmas that leads to the proof of Theorem 5.1.

Definition 5.2 (Root-Oriented Tree): A root-oriented tree is defined in the same way as the WDAG in Definition 3.4 and Theorem 3.7 but with the further constraints that T has a single sink node (which is a variable node) and that T is a tree when viewed as an undirected graph. Note that the name "root-oriented" is due to the fact that the edges are oriented toward the root of the tree, as shown in Figure 7.

Remark 5.3: Algorithm 2 can also be used to generate the directed weighted forest corresponding to the subset of the

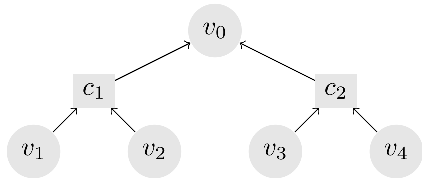

Fig. 7. An example of a root-oriented tree with its root being the variable node v_0.

WDAG consisting of all variable and check nodes that are ancestors[9] *of a given variable node* v. *In this case, the output is a root-oriented tree with its single sink node being the unique replicate of* v.

Definition 5.4 (G_{max}, α_{max}): Let $G = (V, C, E, w, \gamma)$ *be a WDAG. Let* $e_{max} = (v_{max}, c_{max}) = \text{argmax}_{(v,c):w(v,c)\leq 0} |w(v,c)|$ *and let* $\alpha_{max} = |w(v_{max}, c_{max})|$. *Let* $V_{max} = V_{reach} \cup \{v_{max}\}$ *where* V_{reach} *is the set of all variable nodes* $v \in V$ *s.t.* c_{max} *is reachable from* v *in* G *and let* C_{max} *be the set of all check nodes* $c \in C$ *s.t.* c_{max} *is reachable from* c *in* G.[10] *Let* $G_{max} = (V_{max}, C_{max}, E_{max}, w_{max}, \gamma_{max})$ *be the corresponding WDAG.*

Definition 5.5 (Depth of a Variable Node in a Root-Oriented Tree): Let T *be a root-oriented tree with root* v_0. *For any variable node* v *in* T, *the depth of* v *in* T *is defined to be the number of check nodes on the unique directed path from* v *to* v_0 *in* T.

Definition 5.6 (F-Function): Let $G = (V, C, E, w, \gamma)$ *be a WDAG. For any* $S \subseteq V$, *define* $F(S) = \sum_{v\in S}\sum_{c\in N(v):w(v,c)\geq 0} w(v,c)$. *In other words,* $F(S)$ *is the sum of all the "flow" leaving variable nodes in* S *to adjacent check nodes.*

Lemma 5.7: Let $G = (V, C, E, w, \gamma)$ *be a WDAG corresponding to LP decoding of a* (d_v, d_c)*-regular LDPC code (with* $d_v, d_c > 2$*) on the BSC and let* $G_{max} = (V_{max}, C_{max}, E_{max}, w_{max}, \gamma_{max})$ *be the WDAG corresponding to Definition 5.4. Let* $n_{max} = |V_{max}|$ *and* $T = (V', C', E', w', \gamma')$ *be the output of Algorithm 2 on input* G_{max}.[11] *Note that* T *is a root-oriented tree with root* v_{max} *which has a single incoming edge with weight* α_{max} *(by Theorem 4.1). Let* d_{max} *be the maximum depth of a variable node in* T *and for any* $m \in \{0, \ldots, d_{max}\}$, *let* S_m *be the set of all variable nodes in* T *with depth equal to* m. *Moreover, for all* $i \in \{0, \ldots, d_{max}\}$ *and all* $j \in [n_{max}]$, *let* $d_{i,j}$ *denote the number of replicates of variable node* v_j *having depth equal to* i *in* T. *Furthermore, for every* $k \in [d_{i,j}]$, *let* $\Gamma_{i,j,k}$ *be the* γ' *value of the* k*th replicate of* v_j *among those having depth equal to* i *in* T. *Then, for all* $m \in \{1, \ldots, d_{max}\}$, *we have:* (P_m):

$$F(S_m) \geq (d_c - 1)^m \alpha_{max} - \sum_{i=0}^{m-1} (d_c - 1)^{m-i} \sum_{j=1}^{n_{max}} \sum_{k=1}^{d_{i,j}} \Gamma_{i,j,k} \quad (8)$$

Proof of Lemma 5.7: For any $S \subseteq V'$, let $\Delta(S)$ be the set of all $v \in V'$ for which there exist $s \in S$ and a directed path from v to s in T containing exactly one check node. We proceed by induction on m.

Base Case: $m = 1$. We note that $S_1 = \Delta(\{v_{max}\})$ and that v_{max} is the only variable node in T having depth equal to 0 in T. Hence, for the hyperflow to satisfy (6), we should have[12]:

$$\begin{aligned} F(S_1) &\geq (d_c - 1)(\alpha_{max} - \gamma'(v_{max})) \\ &= (d_c - 1)\alpha_{max} - \sum_{i=0}^{0} (d_c - 1)^1 \sum_{j=1}^{n_{max}} \sum_{k=1}^{d_{i,j}} \Gamma_{i,j,k} \end{aligned}$$

Note that the last equality follows from the facts that $d_{0,j} = 1$ if $v_j = v_{max}$ and $d_{0,j} = 0$ otherwise, and that $\Gamma_{i,j,k} = \gamma'(v_{max})$ if $v_j = v_{max}$ and $k = 1$ and $\Gamma_{i,j,k} = 0$ otherwise.

Inductive Step: We need to show that if (P_m) is true for some $1 \leq m \leq d_{max} - 1$, then (P_{m+1}) is also true. Assuming that (P_m) is true, S_m satisfies Equation (8). Since T is a root-oriented tree, $S_{m+1} = \Delta(S_m)$. Hence, for the hyperflow to satisfy (6), we should have:

$$\begin{aligned} F(S_{m+1}) &\geq (d_c - 1)\big(F(S_m) - \sum_{j=1}^{n_{max}} \sum_{k=1}^{d_{m,j}} \Gamma_{m,j,k}\big) \\ &\geq (d_c - 1)[(d_c - 1)^m \alpha_{max} \\ &\quad - \sum_{i=0}^{m-1} (d_c - 1)^{m-i} \sum_{j=1}^{n_{max}} \sum_{k=1}^{d_{i,j}} \Gamma_{i,j,k} - \sum_{j=1}^{n_{max}} \sum_{k=1}^{d_{m,j}} \Gamma_{m,j,k}] \\ &= (d_c - 1)^{m+1} \alpha_{max} \\ &\quad - \sum_{i=0}^{m} (d_c - 1)^{m+1-i} \sum_{j=1}^{n_{max}} \sum_{k=1}^{d_{i,j}} \Gamma_{i,j,k} \end{aligned}$$

□

Definition 5.8 (Depth of a Variable Node in a WDAG With a Single Sink Node): Let $G = (V, C, E, w, \gamma)$ *be a WDAG with a single sink node* $v_0 \in V$ *and let* $v \in V$. *The depth of* v *in* G *is defined to be the minimal number of check nodes on a directed path from* v *to* v_0 *in* G.

Corollary 5.9: Let g_{max} *be the maximum depth of a variable node* $v \in V_{max}$ *in the WDAG* G_{max} *(which has a single sink node* v_{max}*).*[13] *Then,*

$$\alpha_{max} \leq \max_{(T_0,\ldots,T_{g_{max}})\in W} f(T_0, \ldots, T_{g_{max}}) \quad (9)$$

where:

$$f(T_0, \ldots, T_{g_{max}}) = \sum_{i=0}^{g_{max}} \frac{T_i}{(d_c - 1)^i}$$

[9]The notion of "ancestors" here is with respect to the directions of the arrows. For example, in Figure 7, check node c_1 and variable node v_1 are both ancestors of variable node v_0.

[10]Note that $c_{max} \in C_{max}$.

[11]Note that by Definition 5.4, G_{max} has a single sink node v_{max} which has a single incoming edge $e_{max} = (v_{max}, c_{max})$. Hence, part 7 of Theorem 4.1 applies to G_{max}.

[12]In the terminology of (6), we here have that $P_c \geq \alpha_{max} - \gamma'(v_{max})$.

[13]Note that in general $g_{max} \leq d_{max}$ but the two quantities need not be equal.

and W is the set of all tuples $(T_0, \ldots, T_{g_{max}}) \in \mathbb{N}^{g_{max}+1}$ satisfying the following three equations:

$$\sum_{i=0}^{g_{max}} T_i = n_{max} \tag{10}$$

$$T_0 = 1 \tag{11}$$

$$\text{For all } i \in \{0, \ldots, g_{max} - 1\}, \quad T_{i+1} \le (d_c - 1)(d_v - 1)T_i \tag{12}$$

Proof of Corollary 5.9: Setting $m = d_{max}$ in Lemma 5.7 and noting that the leaves of T have no entering flow, we get:

$$\begin{aligned}\sum_{j=1}^{n_{max}} \sum_{k=1}^{d_{d_{max},j}} \Gamma_{d_{max},j,k} &\ge F(S_{d_{max}}) \\ &\ge (d_c - 1)^{d_{max}} \alpha_{max} \\ &\quad - \sum_{i=0}^{d_{max}-1} (d_c - 1)^{d_{max}-i} \sum_{j=1}^{n_{max}} \sum_{k=1}^{d_{i,j}} \Gamma_{i,j,k}\end{aligned}$$

Thus,

$$\alpha_{max} \le \sum_{i=0}^{d_{max}} \frac{1}{(d_c - 1)^i} \sum_{j=1}^{n_{max}} \sum_{k=1}^{d_{i,j}} \Gamma_{i,j,k}$$

Part 6 of Theorem 4.1 implies that for all $v \in V_{max}$, the depth of v in G_{max} is equal to the minimum depth in T of a replicate of v. By parts 3 and 4 of Theorem 4.1, we also have that for all $j \in [n_{max}]$, $\sum_{i=0}^{d_{max}} \sum_{k=1}^{d_{i,j}} \Gamma_{i,j,k} \le 1$ and for all $i \in \{0, \ldots, d_{max}\}$ and all $k \in [d_{i,j}]$, $\Gamma_{i,j,k} \le 1$ and $\{\Gamma_{i,j,k}\}_{i,k}$ all have the same sign. For every $j \in [n_{max}]$, let d_j be the depth of v_j in G_{max} and note that $d_j \le i$ for every $i \in \{0, \ldots, d_{max}\}$ for which there exists $k \in [d_{i,j}]$ s.t. $\Gamma_{i,j,k} \neq 0$. Thus, we get that:

$$\begin{aligned}\alpha_{max} &\le \sum_{i=0}^{d_{max}} \frac{1}{(d_c - 1)^i} \sum_{j=1}^{n_{max}} \sum_{k=1}^{d_{i,j}} |\Gamma_{i,j,k}| \\ &\le \sum_{j=1}^{n_{max}} \frac{1}{(d_c - 1)^{d_j}} \sum_{i=0}^{d_{max}} \sum_{k=1}^{d_{i,j}} |\Gamma_{i,j,k}| \\ &= \sum_{i=0}^{d_{max}} \frac{1}{(d_c - 1)^i} T_i\end{aligned}$$

where the last equality follows from the fact that $\sum_{i=0}^{d_{max}} \sum_{k=1}^{d_{i,j}} |\Gamma_{i,j,k}| = |\sum_{i=0}^{d_{max}} \sum_{k=1}^{d_{i,j}} \Gamma_{i,j,k}| = 1$ for every $j \in [n_{max}]$ with T_i being the number of variable nodes with depth equal to i in G_{max} for every $i \in [d_{max}]$. Note that the notion of depth used here is the one given in Definition 5.8 since G_{max} is a WDAG with a single sink node v_{max}. Since $T_i = 0$ for all $g_{max} < i \le d_{max}$, we get:

$$\alpha_{max} \le \sum_{i=0}^{g_{max}} \frac{1}{(d_c - 1)^i} T_i$$

Equations (10), (11) and (12) follow from the definitions of T_i and g_{max}. □

Lemma 5.10: The RHS of Equation (9) is at most $c \times (n_{max})^{\frac{\ln(d_v-1)}{\ln(d_v-1)+\ln(d_c-1)}}$ for some constant $c > 0$ depending only on d_v.

Proof of Lemma 5.10: Follows from Theorem A.8 in Appendix -C with $\lambda = 1$, $\beta = (d_c - 1)(d_v - 1)$ and $m = n_{max}$. □

Proof of Theorem 5.1: Theorem 5.1 follows from Corollary 5.9 and Lemma 5.10 by noting that $|V_{max}| \le |V|$ since $V_{max} \subseteq V$ and that $\max_{e \in E} |w(e)| = \Omega(\max_{(v,c):w(v,c) \le 0} |w(v,c)|)$ by the hyperflow equation (6). □

We now show that the bound given in Theorem 5.1 is asymptotically tight in the case of (d_v, d_c)-regular LDPC codes.

Theorem 5.11 (Asymptotic Tightness of Theorem 5.1 for (d_v, d_c)-Regular LDPC Codes): There exists an infinite family of (d_v, d_c)-regular Tanner graphs $\{(V_n, C_n, E_n)\}_n$, an infinite family of error patterns $\{\gamma_n\}_n$ and a positive constant c s.t.:

1) *There exists a hyperflow for γ_n on (V_n, C_n, E_n). (Hence, by Theorem 3.7, there exists a WDAG corresponding to a hyperflow for γ_n on (V_n, C_n, E_n).)*
2) *Any WDAG $(V_n, C_n, E_n, w, \gamma_n)$ corresponding to a hyperflow for γ_n on (V_n, C_n, E_n) must have*

$$\max_{e \in E_n} |w(e)| \ge c n^{\frac{\ln(d_v-1)}{\ln(d_v-1)+\ln(d_c-1)}}$$

Proof of Theorem 5.11: See Appendix -D. □

VI. Maximum Weight of an Edge in the WDAG of a Spatially Coupled Code on the BSC

The upper bound of Theorem 5.1 holds for (d_v, d_c)-regular LDPC codes. In this section, we derive a similar sublinear (in the block length n) upper bound that holds for spatially coupled codes.

Theorem 6.1 (Maximum Weight of an Edge in a Spatially Coupled Code): Let $G = (V, C, E, w, \gamma)$ be a WDAG corresponding to LP decoding of any code of the $(d_v, d_c = kd_v, L, M)$ spatially coupled ensemble on the BSC. Let $n = (2L+1)M = |V|$ be the block length of the code. Let $\alpha_{max} = \max_{e \in E} |w(e)|$ be the maximum weight of an edge in G. Then,

$$\alpha_{max} \le c n^{\frac{\ln(q)-\ln(d_c-1)}{\ln(q)}} = c n^{1-\epsilon} = o(n) \tag{13}$$

for some constant $c > 0$ depending only on d_v and where $q = d_v(d_c - 1)\frac{(d_v-1)^{d_v}-1}{d_v-2}$ and $0 < \epsilon = \frac{\ln(d_c-1)}{\ln(q)} < 1$.

We now state and prove a series of lemmas that leads to the proof of Theorem 6.1. Note that a central idea in the proof of Section V is that all check nodes being d_c-regular in that case, the flow at every check node is "cut" by a factor of $d_c - 1$. On the other hand, a $(d_v = 3, d_c = 6, L, M)$ spatially coupled code has $2M$ check nodes with degree 2 and the flow is preserved at such check nodes. To show that even in this case, the maximum weight of an edge is sublinear in the block length, we argue that a check node that is not d_c-regular should have a d_c-regular check node that is "close by" in the WDAG. To simplify the argument, we first "clean" the WDAG of the spatially coupled code to obtain a "reduced WDAG" with all check nodes having either degree d_c or degree 2. We also use a notion of "regular check depth" which is the same as the notion of depth of Section 6.1 except that only d_c-regular check nodes are now counted.

Definition 6.2 (Reduced WDAG): Let $G = (V, C, E, w, \gamma)$ *be a WDAG and* $G_{max} = (V_{max}, C_{max}, E_{max}, w_{max}, \gamma_{max})$ *be the WDAG corresponding to Definition 5.4. The reduced WDAG* G_r *of* G_{max} *is obtained by processing* G_{max} *as follows so that each check node has either degree* d_c *or degree 2:*

1) *For every check node c of* G_r *with spatial index*[14] $< (-L+\hat{d}_v)$*, we remove all the incoming edges to c except one that comes from a parent*[15] *of c having maximal spatial index.*
2) *For every check node c of* T' *with spatial index* $>(L - \hat{d}_v)$*, we remove all the incoming edges to c except for one edge that comes from a parent of c having minimal spatial index.*
3) *We keep only the variable nodes* v *s.t.* v_{max} *is still reachable from* v *and the check nodes c s.t.* v_{max} *is still reachable from c.*

Note that in steps 1 and 2 above, the check nodes of G_r *are considered in an arbitrary order.*

Definition 6.3 (Reduced Tree): A reduced tree with root v_0 *is a root-oriented tree with root* v_0 *and where every check node has either degree* d_c *or degree 2.*

Note that if we run Algorithm 2 on a reduced WDAG, the output will be a reduced tree.

Definition 6.4 (Regular Check Depth of a Variable Node in a Reduced Tree): Let T be a reduced tree with root v_0*. For any variable node* v *of T, the regular check depth of* v *in T is the number of* d_c*-regular check nodes on the directed path from* v *to* v_0 *in T.*

Lemma 6.5: Let $G = (V, C, E, w, \gamma)$ *be a WDAG corresponding to LP decoding of a spatially coupled code on the BSC,* $G_{max} = (V_{max}, C_{max}, E_{max}, w_{max}, \gamma_{max})$ *be the WDAG corresponding to Definition 5.4,* $G_r = (V_r, C_r, E_r, w_r, \gamma_r)$ *be the reduced WDAG corresponding to* G_{max} *and* $T = (V_r', C_r', E_r', w_r', \gamma_r')$ *be the output of Algorithm 2 on input* G_r*. Let* $n_r = |V_r|$*. Note that T is a reduced tree with root* v_{max} *which has a single incoming edge with weight* α_{max} *(by Theorem 4.1). Let* r_{max} *be the maximum regular check depth in T of a variable node* $v \in V_r'$*. For all* $i \in \{0, \ldots, r_{max}\}$ *and all* $j \in [n_r]$*, let* $y_{i,j}$ *be the number of replicates of variable node* v_j *having regular check depth equal to i in T. Moreover, for all* $k \in [y_{i,j}]$*, let* $\Gamma_{i,j,k}$ *denote the* γ_r' *value of the kth replicate of* v_j *among those having regular check depth equal to i in T. Then, for all* $m \in \{1, \ldots, r_{max}\}$*, we have* (P_m)*: There exists* $U_m \subseteq V_r'$ *consisting of variable nodes having regular check depth m in T and s.t. all variable nodes of T having regular check depth between* $m+1$ *and* r_{max} *(inclusive) are ancestors of* U_m *in T and s.t.:*

$$F(U_m) \geq (d_c - 1)^m \alpha_{max} - \sum_{i=0}^{m-1} (d_c - 1)^{m-i} \sum_{j=1}^{n_r} \sum_{k=1}^{y_{i,j}} \Gamma_{i,j,k} \quad (14)$$

Proof of Lemma 6.5: For any $S \subseteq V_r'$, let $\Delta(S)$ be the set of all $v \in V_r'$ for which there exist $s \in S$ and a directed path from v to s in T with the child of v on this path being the unique d_c-regular check node on the path.[16] We proceed by induction on m.

Base Case: $m = 1$. Let $U_1 = \Delta(\{v_{max}\})$. Note that the ancestors of v_{max} (inlcuding v_{max}) that are proper descendants of nodes in U_1 are exactly those variable nodes having regular check depth equal to 0 in T. Hence, for the hyperflow to satisfy Equation (6), we should have:

$$\begin{aligned} F(U_1) &\geq (d_c - 1)\Big(\alpha_{max} - \sum_{j=1}^{n_r} \sum_{k=1}^{y_{0,j}} \Gamma_{0,j,k}\Big) \\ &= (d_c - 1)^1 \alpha_{max} - \sum_{i=0}^{0} (d_c - 1)^1 \sum_{j=1}^{n_r} \sum_{k=1}^{y_{i,j}} \Gamma_{i,j,k} \end{aligned}$$

Inductive Step: We need to show that if (P_m) is true for some $1 \leq m \leq (r_{max} - 1)$ then (P_{m+1}) is also true. Assuming that (P_m) is true, there exists $U_m \subseteq V_r'$ that satisfies Equation (14) and s.t. U_m consists of variable nodes having regular check depth m in T, and all variable nodes of T with regular check depth between $m+1$ and r_{max} (inclusive) are ancestors of U_m in T. Let $U_{m+1} = \Delta(U_m)$. Note that the variable nodes that are ancestors of nodes in U_m and proper descendants of nodes in U_{m+1} are exactly those having regular check depth equal to m in T. Hence, for the hyperflow to satisfy Equation (6), we should have:

$$\begin{aligned} F(U_{m+1}) &\geq (d_c - 1)\Big(F(U_m) - \sum_{j=1}^{n_r} \sum_{k=1}^{y_{m,j}} \Gamma_{m,j,k}\Big) \\ &\geq (d_c - 1)[(d_c - 1)^m \alpha_{max} \\ &\quad - \sum_{i=0}^{m-1} (d_c - 1)^{m-i} \sum_{j=1}^{n_r} \sum_{k=1}^{y_{i,j}} \Gamma_{i,j,k} - \sum_{j=1}^{n_r} \sum_{k=1}^{y_{m,j}} \Gamma_{m,j,k}] \\ &= (d_c - 1)^{m+1} \alpha_{max} \\ &\quad - \sum_{i=0}^{m-1} (d_c - 1)^{m+1-i} \sum_{j=1}^{n_r} \sum_{k=1}^{y_{i,j}} \Gamma_{i,j,k} \\ &\quad - (d_c - 1) \sum_{j=1}^{n_r} \sum_{k=1}^{y_{m,j}} \Gamma_{m,j,k} \\ &= (d_c - 1)^{m+1} \alpha_{max} \\ &\quad - \sum_{i=0}^{m} (d_c - 1)^{m+1-i} \sum_{j=1}^{n_r} \sum_{k=1}^{y_{i,j}} \Gamma_{i,j,k} \end{aligned}$$

Definition 6.6 (Regular Check Depth of a Variable Node in a Reduced WDAG): Let G_r *be a reduced WDAG with its single sink node denoted by* v_0*. For any variable node* v *of* G_r*, the regular check depth of* v *in* G_r *is the minimum number of* d_c*-regular check nodes on a directed path from* v *to* v_0 *in* G_r*.*

Lemma 6.7: Let G_r *be a reduced WDAG and* z_{max} *be the maximum regular check depth of a variable node in* G_r*. For all* $i \in \{0, \ldots, z_{max}\}$*, let* T_i *be the number of variable nodes in* G_r *with regular check depth equal to i. Then, for all*

[14] The notion of "spatial index" used here is the one from Definition 2.1.

[15] The notion of "parent" of a node is the one induced by the direction of the edges of G_r.

[16] Again, the notion of "child" here is the one induced by the direction of the edges of T.

$i \in \{0, \ldots, z_{max} - 1\}$*:*

$$T_{i+1} \le qT_i$$

where $q = d_v(d_c - 1)\frac{(d_v-1)^{d_v}-1}{d_v-2}$*. Moreover,*

$$T_0 \le 1 + \frac{(d_v - 1)^{d_v-1} - 1}{d_v - 2} = q_0$$

Proof of Lemma 6.7: If, for any $i \in \{0, \ldots, z_{max}\}$, we let W_i be the set of all variable nodes in G_r with regular check depth equal to i, then $T_i = |W_i|$. Fix $i \in \{0, \ldots, z_{max} - 1\}$. For a variable node v of G_r, define $\Delta'(v)$ to be the set of all variable nodes v_0 in G_r s.t. there exists a directed path $\mathcal{P}$ from v_0 to v in G_r s.t. the parent of v on $\mathcal{P}$ is the only d_c-regular check node on $\mathcal{P}$. Note that for every variable node $u \in W_{i+1}$, there exists a variable node $v \in W_i$ s.t. $u \in \Delta'(v)$. Thus, $W_{i+1} \subseteq \bigcup_{v \in W_i} \Delta'(v)$ which implies that

$$|W_{i+1}| \le |W_i| \times \max_{v \in W_i} |\Delta'(v)| \le |W_i| \times \max_{v \in V_r} |\Delta'(v)|$$

where V_r is the set of all variable nodes of G_r. We now show that for every $v \in V_r$, $|\Delta'(v)| \le q$. Fix $v \in V_r$. We claim that for all $u \in \Delta'(v)$, there exists a directed path from u to v in G_r containing a single d_c-regular check node which is the parent of v on this path and at most $(d_v - 1)$ 2-regular check nodes. To show this, let $\mathcal{P}$ be a directed path from u to v in G_r containing no d_c-regular check nodes other than the parent of v on this path. If $\mathcal{P}$ does not contain any 2-regular check nodes, then the needed property holds. If $\mathcal{P}$ contains at least one 2-regular check node, then,

$$\mathcal{P} : u \rightsquigarrow c_1 \rightsquigarrow v_1 \rightsquigarrow c_2 \rightsquigarrow v_2 \rightsquigarrow \cdots \rightsquigarrow c_l \rightsquigarrow v_l \rightsquigarrow c_* \rightsquigarrow v \tag{15}$$

where l is a positive integer, $c_1, c_2, \ldots, c_l$ are 2-regular check nodes of G_r, c_* is a d_c-regular check node of G_r and $v_1, v_2, \ldots, v_l$ are variable nodes of G_r. For any check node c, we denote by $si(c)$ the spatial index of c. Since c_1 is 2-regular, its spatial index $si(c_1)$ is either in the interval $[-L - \hat{d}_v : -L + \hat{d}_v - 1]$ or in the interval $[L - \hat{d}_v + 1 : L + \hat{d}_v]$. Without loss of generality, assume that $si(c_1) \in [L - \hat{d}_v + 1 : L + \hat{d}_v]$. For any $i \in \{0, \ldots, l - 1\}$, Definition 6.2 implies that v_i is at a minimal position w.r.t. c_{i+1}. By Definition 2.1, if variable node v is at a minimal position w.r.t. check node c, then c is at a maximal position w.r.t. v. So for any $i \in \{0, \ldots, l-1\}$, c_{i+1} is at a maximal position w.r.t v_i and thus $si(c_i) \le si(c_{i+1})$. By condition 5 of Definition 2.1, variable node v_i is not connected to two check nodes at the same position, which implies that $si(c_i) \ne si(c_{i+1})$ for all $i \in \{0, \ldots, l-1\}$. So we conclude that $si(c_i) < si(c_{i+1})$ for all $i \in \{0, \ldots, l - 1\}$. Therefore,

$$L - \hat{d}_v + 1 \le si(c_1) < si(c_2) < \cdots < si(c_l) \le L + \hat{d}_v$$

Hence, $l \le 2\hat{d}_v = d_v - 1$. So $\mathcal{P}$ satisfies the needed property. For all $i \in [d_v - 1]$, let n_i be the number of variable nodes u in G_r for which the smallest integer l for which Equation (15) holds is $l = i$. Also, let n_0 be the number of variable nodes u in G_r for which there exists a path $\mathcal{P}$ of the form

$$\mathcal{P} : u \rightsquigarrow c_* \rightsquigarrow v \tag{16}$$

where c_* is a d_c-regular check node of G_r. Since in Equation (16) v has at most d_v neighbors in G_r and c_* is d_c-regular, $n_0 \le d_v(d_c - 1)$. Considering Equation (15) with $l = 1$, we note that v_1 has at most d_v neighbors in G_r and c_1 is 2-regular. Thus, $n_1 \le d_v(d_c-1)(d_v-1)$. Note that if u is a variable node in G_r for which the smallest integer l for which Equation (15) holds is $l = i+1$ (where $i \in [d_v-2]$), then there exists a path $\mathcal{P}$ that satisfies Equation (15) with v_1 being a variable node in G_r for which the smallest integer l for which Equation (15) holds is $l = i$. Since for every $l \in [d_v - 1]$ and every $i \in [l]$, v_i has at most d_v neighbors in G_r and c_i is 2-regular, we have that $n_{i+1} \le (d_v - 1)n_i$ for all $i \in [d_v - 2]$. By induction on i, we get that $n_i \le d_v(d_c - 1)(d_v - 1)^i$ for all $i \in [d_v - 1]$. Thus,

$$\begin{aligned} |\Delta'(v)| &= \sum_{i=0}^{d_v-1} n_i \\ &\le \sum_{i=0}^{d_v-1} d_v(d_c - 1)(d_v - 1)^i \\ &= d_v(d_c - 1)\frac{(d_v - 1)^{d_v} - 1}{d_v - 2} \\ &= q \end{aligned}$$

To show that $T_0 \le q_0$, note that $u \in W_0$ if and only if there exists a directed path from u to v_{max} in G_r containing only 2-regular check nodes. An analogous argument to the above implies that

$$T_0 \le 1 + \sum_{i=1}^{d_v-1} (d_v - 1)^{i-1} \le 1 + \frac{(d_v - 1)^{d_v-1} - 1}{d_v - 2} = q_0$$

□

Corollary 6.8: Let G_r be the WDAG (with a single sink node) given in Lemma 6.5 and z_{max} be the maximum regular check depth of a variable node in G_r.[17] *Then,*

$$\alpha_{max} \le \max_{(T_0,\ldots,T_{z_{max}}) \in W} f(T_0, \ldots, T_{z_{max}}) \tag{17}$$

where:

$$f(T_0, \ldots, T_{z_{max}}) = \sum_{i=0}^{z_{max}} \frac{T_i}{(d_c - 1)^i}$$

and W is the set of all tuples $(T_0, \ldots, T_{z_{max}}) \in \mathbb{N}^{z_{max}+1}$ satisfying the following three equations:

$$\sum_{i=0}^{z_{max}} T_i = n_r \tag{18}$$

$$T_0 \le q_0 \tag{19}$$

$$\text{For all } i \in \{0, \ldots, z_{max} - 1\}, \; T_{i+1} \le qT_i \tag{20}$$

where $q = d_v(d_c - 1)\frac{(d_v-1)^{d_v}-1}{d_v-2}$ *and* $q_0 = 1 + \frac{(d_v-1)^{d_v-1}-1}{d_v-2}$.

Proof of Corollary 6.8: The proof is similar to that of Corollary 5.9. Setting $m = r_{max}$ in Lemma 6.5 and noting

[17]Note that in general $z_{max} \le r_{max}$ but the two quantities need not be equal.

that the leaves of T have no entering flow, we get:

$$\sum_{j=1}^{n_r}\sum_{k=1}^{y_{r_{max},j}} \Gamma_{r_{max},j,k} \geq F(U_{r_{max}}) \geq (d_c-1)^{r_{max}}\alpha_{max} - \sum_{i=0}^{r_{max}-1}(d_c-1)^{r_{max}-i}\sum_{j=1}^{n_r}\sum_{k=1}^{y_{i,j}}\Gamma_{i,j,k}$$

Thus,

$$\alpha_{max} \leq \sum_{i=0}^{r_{max}} \frac{1}{(d_c-1)^i}\sum_{j=1}^{n_r}\sum_{k=1}^{y_{i,j}}\Gamma_{i,j,k}$$

Part 6 of Theorem 4.1 implies that for every $v \in V_r$, the regular check depth of v in G_r is equal to the minimum regular check depth in T of a replicate of v. By parts 3 and 4 of Theorem 4.1, we also have that for all $j \in [n_r]$, $\sum_{i=0}^{r_{max}}\sum_{k=1}^{y_{i,j}}\Gamma_{i,j,k} \leq 1$ and for all $i \in \{0,\ldots,r_{max}\}$ and all $k \in [y_{i,j}]$, $\Gamma_{i,j,k} \leq 1$ and $\{\Gamma_{i,j,k}\}_{i,k}$ all have the same sign. Thus, we get that:

$$\alpha_{max} \leq \sum_{i=0}^{r_{max}} \frac{1}{(d_c-1)^i} T_i$$

where for every $i \in \{0,\ldots,r_{max}\}$, T_i is the number of variable nodes with regular check depth equal to i in G_r. Since $T_i = 0$ for all $z_{max} < i \leq r_{max}$, we get that:

$$\alpha_{max} \leq \sum_{i=0}^{z_{max}} \frac{1}{(d_c-1)^i} T_i$$

By the definitions of T_i and z_{max}, $\sum_{i=0}^{z_{max}} T_i = n_r$. The facts that $T_{i+1} \leq qT_i$ for all $i \in \{0,\ldots,z_{max}-1\}$ and $T_0 \leq q_0$ follow from Lemma 6.7. □

Lemma 6.9: The RHS of (17) is $< c \times n_r^{1-\epsilon}$ for some constant $c > 0$ depending only on d_v and where $0 < \epsilon = \frac{\ln(d_c-1)}{\ln(q)} < 1$.

Proof of Lemma 6.9: Let $c = q_0 \frac{\left(\frac{q}{d_c-1}\right)^2}{\frac{q}{d_c-1}-1}$. If $n_r \geq q_0$, the claim follows from Theorem A.8 in Appendix -C with $\lambda = q_0$, $\beta = q$ and $m = n_r$. If $n_r < q_0$, then the RHS of (17) is at most $n_r < q_0 < c$, so the claim is also true. □

Proof of Theorem 6.1: Theorem 6.1 follows from Corollary 6.8 and Lemma 6.9 by noting that $|V_r| \leq |V|$ since $V_r \subseteq V$ and that $\max_{e\in E} |w(e)| = \Omega(\max_{(v,c):w(v,c)\leq 0} |w(v,c)|)$ by the hyperflow equation (6). □

VII. Relation Between LP Decoding on a Tail-Biting Graph Cover Code and on a Derived Spatially Coupled Code

Definition 7.1 (Special Variable Nodes and Extra Flow): Let ζ be a tail-biting graph cover code and ζ' be a fixed element of $\mathcal{D}(\zeta)$.[18] *Then, the "special variable nodes" of ζ are all those variable nodes that appear in ζ but not in ζ'. If η is an error pattern on ζ, a dual witness for η on ζ with "extra flow" f is a dual witness satisfying Definition 3.1 with the exception that for every special variable node v, Equation 4 is replaced by*

$$\sum_{c\in N(v):w(v,c)>0} w(v,c) < \sum_{c\in N(v):w(v,c)\leq 0} (-w(v,c)) + \gamma(v) + f \tag{21}$$

where γ is the log-likelihood ratio corresponding to η.

Lemma 7.2: Let ζ be a $(d_v, d_c = kd_v, L, M)$ tail-biting graph cover code and let ζ' be a be a fixed element of $\mathcal{D}(\zeta)$. Let $n = (2L+1)M$ be the block length of ζ and consider transmission over the BSC. Assume that there is an $\alpha(n)$ s.t., for any error pattern η' on ζ', the existence of a dual witness for η' on ζ' implies the existence of a dual witness for η' on ζ' with maximum edge weight $< \alpha(n)$.
Then, for any error pattern η' on ζ' and any extension η of η' into an error pattern on ζ, the existence of a dual witness for η' on ζ' is equivalent to the existence of a dual witness for η on ζ with the special variable nodes having an extra flow of $d_v\alpha(n)+1$.

Proof of Lemma 7.2: First, we prove the forward direction of the equivalence. Assume that there exists a dual witness for η' on ζ'. Then, there exists a dual witness for η' on ζ' and with maximum edge weight $< \alpha(n)$. This implies the existence of a dual witness for η on ζ with the special variable nodes being source nodes and having an extra flow of $d_v\alpha(n)+1$.
The reverse direction follows from the fact that given a dual witness for η on ζ, we can get a dual witness for η' on ζ' by repeatedly removing the special variable nodes. The WDAG satisfies the LP constraints after each step since every check node in ζ' has degree ≥ 2. □

Corollary 7.3 (Relation Between LP Decoding on a Tail-Biting Graph Cover Code and on a Derived Spatially Coupled Code): Let ζ be a $(d_v, d_c = kd_v, L, M)$ tail-biting graph cover code and let ζ' be a be a fixed element of $\mathcal{D}(\zeta)$. Let $n = (2L+1)M$ be the block length of ζ and consider transmission over the BSC. Then, for any error pattern η' on ζ' and any extension η of η' into an error pattern on ζ, the existence of a dual witness for η' on ζ' is equivalent to the existence of a dual witness for η on ζ with the special variable nodes having an extra flow of $d_v cn^{1-\epsilon}+1$ for some $c > 0$ and $0 < \epsilon < 1$ given in Theorem 6.1.

Proof of Corollary 7.3: By Theorem 6.1, the existence of a dual witness for η' on ζ' is equivalent to the existence of a dual witness for η' on ζ' and with maximum edge weight $< cn^{1-\epsilon}$ for some $c > 0$. Plugging this expression in Lemma 7.2, we get the statement of Corollary 7.3. □

VIII. Interplay Between Crossover Probability and LP Excess

In this section, we show that if the probability of LP decoding success is large on some BSC, then if we slightly decrease the crossover probability of the BSC, we can find a dual witness with a non-negligible "gap" in the inequalities (4) with high probability.

Theorem 8.1 (Interplay Between Crossover Probability and LP Excess): Let ζ be a binary linear code with Tanner graph (V, C, E) where $V = \{v_1, \cdots, v_n\}$. Let $\epsilon, \delta > 0$ and

[18]Here, $\mathcal{D}(\zeta)$ refers to Definition 2.3.

$\epsilon' = \epsilon + (1-\epsilon)\delta$. Assume that $\epsilon, \epsilon', \delta < 1$. Let $q_{\epsilon'}$ be the probability of LP decoding error on the ϵ'-BSC. For every error pattern $x \in \{0,1\}^n$, if $G = (V, C, E, w, \gamma)$ is a WDAG corresponding to a dual witness for x, let $f(w) \in \mathbb{R}^n$ be defined by

$$f_i(w) = \sum_{\substack{c \in N(v_i):\\ w(v_i,c)>0}} w(v_i, c) - \sum_{\substack{c \in N(v_i):\\ w(v_i,c)\leq 0}} (-w(v_i, c)) = \sum_{c \in N(v_i)} w(v_i, c) \quad (22)$$

for all $i \in [n]$. Then, for $x \sim \mathrm{Ber}(\epsilon, n)$, we have:

$$\Pr_x\{\exists \text{ a dual witness } w \text{ for } x \text{ s.t. } f_i(w) < \gamma(v_i) - \tfrac{\delta}{2}, \forall i\} \geq 1 - \tfrac{2q_{\epsilon'}}{\delta}$$

In other words, if we let $\gamma(v_i) - f_i(w)$ be the "LP excess" on variable node i, then the probability (over the ϵ-BSC) that there exists a dual witness with LP excess at least $\delta/2$ on all the variable nodes is at least $1 - \frac{2q_{\epsilon'}}{\delta}$.

Proof of Theorem 8.1: Decompose the ϵ'-BSC into the bitwise OR of the ϵ-BSC and the δ-BSC as follows. Let $x \sim \mathrm{Ber}(\epsilon, n)$, $e'' \sim \mathrm{Ber}(\delta, n)$ and $e = x \vee e''$. Hence, $e \sim \mathrm{Ber}(\epsilon', n)$. For every $x \in \{0,1\}^n$, we will construct a dual witness w^x with excess $\delta/2$ on all variable nodes by averaging and scaling the dual witnesses of $x \vee e''$ where $e'' \sim \mathrm{Ber}(\delta, n)$. More precisely, for every $x \in \{0,1\}^n$, let $w^x = \frac{(1+\frac{\delta}{2})}{(1-\frac{\delta}{2})} E_{e''\sim \mathrm{Ber}(\delta,n)}\{v^{x\vee e''}\}$ where v^x is an arbitrary dual witness for x if x has one and v^x is the zero vector otherwise. Note that w^x always satisfies the check node constraints, i.e. for any $x \in \{0,1\}^n$, any $c \in C$ and any $v, v' \in V$, we have $w^x(v, c) + w^x(v', c) \geq 0$. We now show that, with probability at least $1 - \frac{2q_{\epsilon'}}{\delta}$ over $x \sim \mathrm{Ber}(\epsilon, n)$, w^x satisfies (4) with LP excess at least $\delta/2$ on all variable nodes. For any weight function $w : V \times C \to \mathbb{R}$ on the Tanner graph (V, C, E), we define $f(w)$ by Equation (22). For every $x \in \{0,1\}^n$, define the event $L^x = \{x \text{ has a dual witness}\}$ and define $\tilde{x}$ by $\tilde{x}_i = (-1)^{x_i}$ for all $i \in [n]$. We have that:

$$\begin{aligned}
f(w^x) &= \frac{(1+\frac{\delta}{2})}{(1-\frac{\delta}{2})} E_{e''\sim\mathrm{Ber}(\delta,n)}\{f(w^{x\vee e''})\} \\
&= \frac{(1+\frac{\delta}{2})}{(1-\frac{\delta}{2})} \Big(E_{e''}\{f(w^{x\vee e''}) | L^{x\vee e''}\} \Pr_{e''}\{L^{x\vee e''}\} \\
&\qquad + E_{e''}\{f(w^{x\vee e''}) | \overline{L^{x\vee e''}}\} \Pr_{e''}\{\overline{L^{x\vee e''}}\} \Big) \\
&\overset{(a)}{=} \frac{(1+\frac{\delta}{2})}{(1-\frac{\delta}{2})} E_{e''}\{f(w^{x\vee e''}) | L^{x\vee e''}\} \Pr_{e''}\{L^{x\vee e''}\} \\
&\overset{(b)}{\leq} \frac{(1+\frac{\delta}{2})}{(1-\frac{\delta}{2})} E_{e''}\{\widetilde{x \vee e''} | L^{x\vee e''}\} \Pr_{e''}\{L^{x\vee e''}\} \\
&= \frac{(1+\frac{\delta}{2})}{(1-\frac{\delta}{2})} \Big(E_{e''}\{\widetilde{x \vee e''}\} - E_{e''}\{\widetilde{x \vee e''} | \overline{L^{x\vee e''}}\} \times \phi_x \Big)
\end{aligned}$$

where (a) follows from $E_{e''\sim\mathrm{Ber}(\delta,n)}\{f(w^{x\vee e''}) | \overline{L^{x\vee e''}}\} = 0$, (b) follows from Equation (4) and $\phi_x := \Pr_{e''\sim\mathrm{Ber}(\delta,n)}\left\{\overline{L^{x\vee e''}}\right\}$.

Note that for $e'' \sim \mathrm{Ber}(\delta, n)$ and for every $i \in [n]$, we have

$$(E_{e''}\{\widetilde{x \vee e''}\})_i = \begin{cases} -1 & \text{if } x_i = 1 \\ \delta(-1) + (1-\delta)(1) = 1 - 2\delta & \text{if } x_i = 0 \end{cases}$$

Moreover, $\underset{\sim}{E_{e''\sim\mathrm{Ber}(\delta,n)}}\{\widetilde{x \vee e''} | \overline{L^{x\vee e''}}\} \geq -1$ since every coordinate of $\widetilde{x \vee e''}$ is ≥ -1. Therefore,

$$f_i(w^x) \leq \begin{cases} \frac{(1+\frac{\delta}{2})}{(1-\frac{\delta}{2})}(-1 + \phi_x) & \text{if } x_i = 1 \\ \frac{(1+\frac{\delta}{2})}{(1-\frac{\delta}{2})}(1 - 2\delta + \phi_x) & \text{if } x_i = 0 \end{cases}$$

We now find an upper bound on ϕ_x. Note that ϕ_x is a non-negative random variable with mean

$$\begin{aligned}
E_{x\sim\mathrm{Ber}(\epsilon,n)}\{\phi_x\} &= E_{x\sim\mathrm{Ber}(\epsilon,n)}\big\{\Pr_{e''\sim\mathrm{Ber}(\delta,n)}\{\overline{L^{x\vee e''}}\}\big\} \\
&= \Pr_{x\sim\mathrm{Ber}(\epsilon,n), e''\sim\mathrm{Ber}(\delta,n)}\left\{\overline{L^{x\vee e''}}\right\} \\
&= \Pr_{e\sim\mathrm{Ber}(\epsilon',n)}\left\{\overline{L^{e}}\right\} \\
&= q_{\epsilon'} \quad \text{(by Theorem 3.2)}
\end{aligned}$$

By Markov's inequality,

$$\Pr_{x\sim\mathrm{Ber}(\epsilon,n)}\{\phi_x \geq \frac{\delta}{2}\} \leq \frac{E_{x\sim\mathrm{Ber}(\epsilon,n)}\{\phi_x\}}{\frac{\delta}{2}} = \frac{2q_{\epsilon'}}{\delta}$$

Thus, the probability over $x \sim \mathrm{Ber}(\epsilon, n)$ that for all $i \in [n]$,

$$f_i(w^x) < \frac{(1+\frac{\delta}{2})}{(1-\frac{\delta}{2})}(-1 + \frac{\delta}{2}) \quad \text{if } x_i = 1$$

and

$$f_i(w^x) < \frac{(1+\frac{\delta}{2})}{(1-\frac{\delta}{2})}(1 - \frac{3\delta}{2}) \quad \text{if } x_i = 0$$

is at least

$$\Pr_{x\sim\mathrm{Ber}(\epsilon,n)}\{\phi_x < \frac{\delta}{2}\} = 1 - \Pr_{x\sim\mathrm{Ber}(\epsilon,n)}\{\phi_x \geq \frac{\delta}{2}\} \geq 1 - \frac{2q_{\epsilon'}}{\delta}$$

Note that for all $0 \leq \delta < 1$, we have that

$$\frac{(1+\frac{\delta}{2})}{(1-\frac{\delta}{2})}(1 - \frac{3\delta}{2}) \leq 1 - \frac{\delta}{2}$$

Thus, the probability over $x \sim \mathrm{Ber}(\epsilon, n)$ that $f_i(w^x) < (-1)^{x_i} - \frac{\delta}{2}$ for all $i \in [n]$, is at least $1 - \frac{2q_{\epsilon'}}{\delta}$. So we conclude that for $x \sim \mathrm{Ber}(\epsilon, n)$, we have

$$\Pr_x\{\exists \text{ a dual witness } w \text{ for } x \text{ s.t. } f_i(w) < \gamma(v_i) - \tfrac{\delta}{2}, \forall i\} \geq 1 - \tfrac{2q_{\epsilon'}}{\delta}$$

□

IX. Proof of Main Result: $\xi_{GC} = \xi_{SC}$

In this section, we use the results of Sections VI, VII and VIII to prove the main result of the paper which is restated below.

Theorem 9.1 (Main Result: $\xi_{GC} = \xi_{SC}$): Let Γ_{GC} be a $(d_v, d_c = kd_v, L, M)$ tail-biting graph cover ensemble with d_v an odd integer and M divisible by k. Let Γ_{SC} be the $(d_v, d_c = kd_v, L - \hat{d}_v, M)$ spatially coupled ensemble which is sampled by choosing a tail-biting graph cover code $\zeta \sim \Gamma_{GC}$ and *returning a element of $\mathcal{D}(\zeta)$ chosen uniformly at random*[19]. *Denote by ξ_{GC} and ξ_{SC} the respective LP threholds of Γ_{GC} and Γ_{SC} on the BSC. Then, there exists $\nu > 0$ depending only on d_v and d_c s.t. if $M = o(L^\nu)$ and Γ_{SC} satisfies the property that for any constant $\Delta > 0$,*

$$\Pr_{\substack{\zeta' \sim \Gamma_{SC} \\ (\xi_{SC}-\Delta)\text{-}BSC}}[LP \text{ decoding error on } \zeta'] = o(\frac{1}{L^2}) \quad (23)$$

then, $\xi_{GC} = \xi_{SC}$.

Lemma 9.2: Assume that the ensemble Γ_{SC} satisfies the property (23) for every constant $\Delta > 0$. Then, for all constants $\Delta_1, \Delta_2, \alpha, \beta > 0$, there exists a tail-biting graph cover code $\zeta \in \Gamma_{GC}$, with derived spatially coupled codes $\zeta'_{-L}, \ldots, \zeta'_L$, satisfying the following two properties for sufficiently large L:

1) $\Pr_{(\xi_{GC}+\Delta_2)\text{-}BSC}$[*LP decoding success on* ζ] $\leq \alpha$.
2) *For all* $i \in [-L : L]$, $\Pr_{(\xi_{SC}-\Delta_1)\text{-}BSC}$[*LP decoding error on* ζ'_i] $\leq \beta/(2L+1)$.

Proof of Lemma 9.2: Note that a random code $\zeta \sim \Gamma_{GC}$ satisfies the 2 properties above with high probability:

$$\begin{aligned}
&\Pr_{\zeta \sim \Gamma_{GC}}\Big[\Pr_{(\xi_{GC}+\Delta_2)\text{-}BSC}[\text{Success on } \zeta] > \alpha \text{ or } \exists i \\
&\in [-L:L] \text{ s.t. } \Pr_{(\xi_{SC}-\Delta_1)\text{-}BSC}[\text{Error on } \zeta'_i] > \beta(2L+1)\Big] \\
&\leq \frac{1}{\alpha}\Pr_{\substack{\zeta \sim \Gamma_{GC} \\ (\xi_{GC}+\Delta_2)\text{-}BSC}}[\text{LP decoding success on } \zeta] \\
&\quad + \frac{(2L+1)^2}{\beta}\Pr_{\substack{\zeta' \sim \Gamma_{SC} \\ (\xi_{SC}-\Delta_1)\text{-}BSC}}[\text{LP decoding error on } \zeta'] \\
&= o(1)
\end{aligned}$$

Note that the inequality above follows from Markov's inequality and the union bound. We conclude that there exists a tail-biting graph cover code $\zeta \in \Gamma_{GC}$ satisfying the 2 properties above. □

Lemma 9.3: $\xi_{GC} \geq \xi_{SC}$

Proof of Lemma 9.3: We proceed by contradiction. Assume that $\xi_{GC} < \xi_{SC}$. Let:

$$\begin{aligned}
\delta &= (\xi_{SC} - \xi_{GC})/2 \\
\eta &= \xi_{SC} - \delta \\
\lambda &= \eta - \delta/2 = \xi_{GC} + \delta/2
\end{aligned}$$

Note that $\eta > \lambda + (1-\lambda)\delta/2$. Let ζ be one of the tail-biting graph cover codes whose existence is guaranteed by Lemma 9.2 with $\Delta_1 = \delta$, $\Delta_2 = \delta/2$ and $\alpha, \beta > 0$ with $\alpha < 1 - 2\beta/\delta$ and let $\zeta'_{-L}, \ldots, \zeta'_L$ be the spatially coupled codes that are derived from ζ. Let μ be an error pattern on ζ and let μ_i be the restriction of μ to ζ'_i for every $i \in [-L : L]$. Define the event:

$$E_1 = \Big\{\forall i \in [-L:L], \exists \text{ a dual witness for } \mu_i \text{ on } \zeta_i' \text{ with excess } \delta/2 \text{ on all variable nodes}\Big\}$$

[19]Here, $\mathcal{D}(\zeta)$ refers to Definition 2.3.

Then,

$$\overline{E_1} = \Big\{\exists i \in [-L:L] \text{ s.t. } \nexists \text{ a dual witness for } \mu_i \text{ on } \zeta_i' \text{ with excess } \delta/2 \text{ on all variable nodes}\Big\}$$

Thus,

$$\begin{aligned}
\Pr_{\lambda\text{-}BSC}\{\overline{E_1}\} &\leq \sum_{i=-L}^{L}\Pr_{\lambda\text{-}BSC}\Big\{\nexists \text{ a dual witness for } \zeta_i' \text{ with excess } \frac{\delta}{2} \text{ on all variable nodes}\Big\} \\
&\overset{(a)}{\leq} \sum_{i=-L}^{L}\frac{2}{\delta}\Pr_{\eta\text{-}BSC}\{\text{LP decoding error on } \zeta_i'\} \\
&\leq \sum_{i=-L}^{L}\frac{2}{\delta}\times\frac{\beta}{2L+1} = \frac{2\beta}{\delta}
\end{aligned}$$

where (a) follows from Theorem 8.1. If event E_1 is true, then by Corollary 7.3, for every $l \in [-L : L]$, there exists a dual witness $\{\tau_{ij}^l \mid i \in V, j \in C\}$ for μ on ζ with the special variable nodes being at positions $[l, l + 2\hat{d}_v - 1]$ and having an extra flow of $d_v c n^{1-\epsilon} + 1$ with $c > 0$ and $\epsilon > 0$ given in Theorem 6.1 and with the non-special variable nodes having excess $\frac{\delta}{2}$. Then, we can construct a dual witness for μ on the tail-biting graph cover code ζ (with no extra flows) by averaging the above $2L+1$ dual witnesses as follows. For every $i \in V$ and every $j \in C$, let:

$$\tau_{ij}^{avg} = \frac{1}{2L+1}\sum_{l=-L}^{L}\tau_{ij}^l$$

We claim that $\{\tau_{ij}^{avg}\}_{i,j}$ forms a dual witness for μ on ζ. In fact, for each $i \in V$, $j \in C$ and $l \in [-L : L]$, $\tau_{ij}^l + \tau_{i'j}^l \geq 0$ which implies that:

$$\tau_{ij}^{avg} + \tau_{i'j}^{avg} = \frac{1}{2L+1}\sum_{l=-L}^{L}(\tau_{ij}^l + \tau_{i'j}^l) \geq 0$$

Moreover, for all $i \in V$, we have that:

$$\begin{aligned}
\sum_{j\in N(i)}\tau_{ij}^{avg} &= \sum_{j\in N(i)}\Big(\frac{1}{2L+1}\sum_{l=-L}^{L}\tau_{ij}^l\Big) \\
&= \frac{1}{2L+1}\sum_{l=-L}^{L}\Big(\sum_{j\in N(i)}\tau_{ij}^l\Big) \\
&< \frac{1}{2L+1}((d_v-1)(d_v c(M(2L+1))^{1-\epsilon} + 1 + \gamma_i) \\
&\quad + (2L+1-(d_v-1))(\gamma_i - \frac{\delta}{2})) \\
&= \gamma_i + (d_v-1)d_v c\frac{(M(2L+1))^{1-\epsilon}}{2L+1} + \frac{(d_v-1)\delta}{2(2L+1)} \\
&\quad + \frac{d_v-1}{2L+1} - \frac{\delta}{2} < \gamma_i
\end{aligned}$$

where the last inequality holds for $M = o(L^{\nu})$, L sufficiently large and $\nu = \epsilon/(1-\epsilon)$. Since

$$\Pr_{\lambda\text{-}BSC}\{\text{LP decoding success on } \zeta\} \geq \Pr_{\lambda\text{-}BSC}\{E_1\} = 1 - \Pr_{\lambda\text{-}BSC}\{\overline{E_1}\}$$

then,

$$\Pr_{\lambda\text{-}BSC}\{\text{LP decoding success on } \zeta\} \geq 1 - \frac{2\beta}{\delta}$$

which contradicts the fact that:

$$\begin{aligned}\Pr_{\lambda\text{-}BSC}&[\text{LP decoding success on } \zeta] \\ &= \Pr_{(\xi_{GC}+\Delta_2)\text{-}BSC}[\text{LP decoding success on } \zeta] \\ &\leq \alpha < 1 - \frac{2\beta}{\delta}\end{aligned}$$

□

Lemma 9.4: $\xi_{GC} \leq \xi_{SC}$

Proof of Lemma 9.4: Let ζ be a tail-biting graph cover code and $D(\zeta)$ be the set of all derived spatially coupled codes of ζ. Let μ be an error pattern on ζ and μ' be the restriction of μ to ζ' for some $\zeta' \in D(\zeta)$. Given a dual witness for μ on ζ, we can get a dual witness for μ' on ζ' by repeatedly removing the special variable nodes of ζ. Note that the dual witness is maintained after each step since every check node in ζ' has degree ≥ 2. So if there is LP decoding success for η on ζ, then for every $\zeta' \in D(\zeta)$, there is LP decoding success for η' on ζ', where η' is the restriction of η to ζ'. Therefore, for every $\epsilon > 0$ and every $\zeta' \in D(\zeta)$, we have that:

$$\Pr_{\epsilon\text{-}BSC}[\text{LP error on } \zeta'] \leq \Pr_{\epsilon\text{-}BSC}[\text{LP error on } \zeta]$$

This implies that for every $\epsilon > 0$, we have that:

$$\Pr_{\substack{\zeta'\sim\Gamma_{SC}\\ \epsilon\text{-}BSC}}[\text{LP error on } \zeta'] \leq \Pr_{\substack{\zeta\sim\Gamma_{GC}\\ \epsilon\text{-}BSC}}[\text{LP error on } \zeta]$$

So we conclude that $\xi_{GC} \leq \xi_{SC}$. □

Proof of Theorem 9.1: Theorem 9.1 follows from Lemma 9.3 and Lemma 9.4. □

X. Open Questions

It was reported by [5] that, based on numerical simulations, spatial coupling does not seem to improve the performance of LP decoding. This lead to the belief that the LP threshold of a spatially coupled ensemble on the BSC is the same as that of the base ensemble, which was the original motivation behind this work. One possible approach to prove this claim is twofold:

1) Show that the LP threshold of the spatially coupled ensemble on the BSC is the same as that of the tail-biting graph cover ensemble.
2) Show that the LP threshold of the tail-biting graph cover ensemble on the BSC is the same as that of the base ensemble.

In this paper, we proved Part 1 of this approach. We leave Part 2 open. While the analogous statement of Part 2 for BP decoding follows from the fact that the base ensemble and the tail-biting graph-cover ensemble have the same local-tree structure, such an argument would fail for the LP decoder which is a global decoder. Since the performance of min-sum is believed to be generally similar to that of LP decoding, an interesting related question is whether there is an improvement in the performance of min-sum under spatial coupling on the BSC, and if not why do min-sum and BP differ so significantly?

Appendix

A. Proof of Theorem 3.2

The goal of this section is to prove Theorem 3.2 which is restated below.

Theorem 3.2 (Existence of a Dual Witness and LP Decoding Success): Let $\mathcal{T} = (V, C, E)$ be a Tanner graph of a binary linear code with block length n and let $\eta \in \{0,1\}^n$ be any error pattern. Then, there is LP decoding success for η on $\mathcal{T}$ if and only if there is a dual witness for η on $\mathcal{T}$.

Note that the "if" part of the statement was proved in [9]. The argument below establishes both directions. We first state some definitions and prove some facts from convex geometry that will be central to the proof of Theorem 3.2.

Definition A.1: Let S be a subset of $\mathbb{R}^n$. The convex hull of S is defined to be $\text{conv}(S) = \{\alpha x + (1-\alpha)y \mid x, y \in S \text{ and } \alpha \in [0,1]\}$. *The conic hull of S is defined to be* $\text{cone}(S) = \{\alpha x + \beta y \mid x, y \in S \text{ and } \alpha, \beta \in \mathbb{R}_{\geq 0}\}$. *The set S is said to be convex if $S = \text{conv}(S)$ and S is said to be a cone if $S = \text{cone}(S)$. Also, S is said to be a convex polyhedron if $S = \{x \in \mathbb{R}^n \mid Ax \geq b\}$ for some matrix $A \in \mathbb{R}^{m\times n}$ and some $b \in \mathbb{R}^n$ and S is said to be a polyhedral cone if S is both a convex polyhedron and a cone. The interior of S is denoted by* $\text{int}(S)$ *and the closure of S is denoted by* $\text{cl}(S)$.
Let K be a polyhedral cone of the form $K = \{x \in \mathbb{R}^n \mid Ax \geq 0\}$ for some matrix $A \in \mathbb{R}^{m\times n}$. For any $x \in K$ s.t. $x \neq 0$, the ray of K in the direction of x is defined to be the set $R(x) = \{\lambda x \mid \lambda \geq 0\}$. A ray $R(x)$ of K is said to be an extreme ray of K if for any $y, z \in \mathbb{R}^n$ and any $\alpha, \beta \geq 0$, $R(x) = \alpha R(y) + \beta R(z)$ implies that $y, z \in R(x)$.

Lemma A.2: If S is a convex subset of $\mathbb{R}^n$, then $\text{int}\big((\mathbb{R}_{\geq 0})^n + S\big) = (\mathbb{R}_{>0})^n + S$.

Proof of Lemma A.2: For all $\alpha \in (\mathbb{R}_{>0})^n + S$, $\alpha = r + s$ where $r \in (\mathbb{R}_{>0})^n$ and $s \in S$. Thus, the ball centered at α and of radius $\min_{i\in[n]} r_i > 0$ is contained in $\big((\mathbb{R}_{\geq 0})^n + S\big)$. Hence, $\alpha \in \text{int}\big((\mathbb{R}_{\geq 0})^n + S\big)$. Therefore, $(\mathbb{R}_{>0})^n + S \subseteq \text{int}\big((\mathbb{R}_{\geq 0})^n + S\big)$.
Conversely, for all $\alpha \in \text{int}\big((\mathbb{R}_{\geq 0})^n + S\big)$, $\alpha = r + s$ where $r \in (\mathbb{R}_{\geq 0})^n$ and $s \in S$. Moreover, since $\alpha \in \text{int}\big((\mathbb{R}_{\geq 0})^n + S\big)$, there exists $u \in (\mathbb{R}_{>0})^n$ s.t. $\alpha + u \in \big((\mathbb{R}_{\geq 0})^n + S\big)$ and $\alpha - u \in \big((\mathbb{R}_{\geq 0})^n + S\big)$. Note that $\alpha + u = r + u + s$ and that $\alpha - u = r' + s'$ for some $r' \in (\mathbb{R}_{\geq 0})^n$ and $s' \in S$. Thus, $\alpha = \frac{(\alpha+u)+(\alpha-u)}{2} = \frac{r+u+r'}{2} + \frac{s+s'}{2} = r'' + s''$ where $r'' = \frac{r+u+r'}{2} \in (\mathbb{R}_{>0})^n$ and $s'' = \frac{s+s'}{2} \in S$ since S is a convex set. Hence, $\text{int}\big((\mathbb{R}_{\geq 0})^n + S\big) \subseteq (\mathbb{R}_{>0})^n + S$.
Therefore, $\text{int}\big((\mathbb{R}_{\geq 0})^n + S\big) = (\mathbb{R}_{>0})^n + S$. □

Lemma A.3: Let $S_1, .., S_p$ be finite subsets of $\mathbb{R}^n$ each containing the zero vector. Then,

$$\text{cone}\Big(\bigcap_{j=1}^{p} \text{conv}(S_j)\Big) = \bigcap_{j=1}^{p} \text{cone}(S_j).$$

Proof of Lemma A.3: Clearly, $\text{cone}\big(\bigcap_{j=1}^{p}\text{conv}(S_j)\big) \subseteq \bigcap_{j=1}^{p}\text{cone}(S_j)$. To prove the other direction, we first note that $0 \in \text{cone}\big(\bigcap_{j=1}^p \text{conv}(S_j)\big)$. For any non-zero $x \in \bigcap_{j=1}^p \text{cone}(S_j)$, we have that for all $j \in [p]$, $x = \sum_{s \in S_j} a_{s,j} s$ where for any $s \in S_j$, $a_{s,j} \geq 0$. Let $j_{max} = \text{argmax}_{j \in [p]} \sum_{s \in S_j} a_{s,j}$. Since $x \neq 0$, $D = \sum_{s \in S_{j_{max}}} a_{s,j_{max}} > 0$. Thus, for any $j \in [p]$, we have $\frac{x}{D} = \sum_{s \in S_j} \big(\frac{a_{s,j}}{D}\big) s + \big(1 - \sum_{s \in S_j} \frac{a_{s,j}}{D}\big) 0$. Since for all $j \in [p]$, $0 \leq \sum_{s \in S_j} a_{s,j} \leq D$ and $0 \in S_j$, we conclude that $\frac{x}{D} \in \text{conv}(S_j)$ for all $j \in [p]$. Hence, $x \in \text{cone}\big(\bigcap_{j=1}^p \text{conv}(S_j)\big)$. Therefore, $\bigcap_{j=1}^p \text{cone}(S_j) \subseteq \text{cone}\big(\bigcap_{j=1}^p \text{conv}(S_j)\big)$. □

Lemma A.4: Let K be a polyhedral cone of the form $K = \{x \in \mathbb{R}^m \mid Ax \geq 0\}$ for some matrix $A \in \mathbb{R}^{l \times m}$ of rank m. For any $x \in K$ s.t. $x \neq 0$, we have

1) *If $R(x)$ is an extreme ray of K, then there exists an $(m-1) \times m$ submatrix A' of A s.t. the rows of A' are linearly independent and $A'x = 0$.*
2) *$K = \text{cone}(R)$ where $R = \bigcup_{\text{extreme rays } R(x) \text{ of } K} R(x)$.*

Proof of Lemma A.4: See Section 8.8 of [21]. □

The following lemma has been used in previous work on LP decoding. We provide a proof for completeness.

Lemma A.5: For all $m \geq 2$, we have that

$$\big\{y \in (\mathbb{R}_{\geq 0})^m \mid \sum_{i=1,\ i \neq i_0}^m y_i \geq y_{i_0}, \forall i_0 \in [m]\big\} = \text{cone}\{z \in \{0,1\}^m \mid wt(z) = 2\}$$

Proof of Lemma A.5: Let

$$K_m = \big\{y \in (\mathbb{R}_{\geq 0})^m \mid \sum_{i=1,\ i \neq i_0}^m y_i \geq y_{i_0}, \forall i_0 \in [m]\big\}$$

and $X_m = \text{cone}\{z \in \{0,1\}^m \mid wt(z) = 2\}$.[20] Clearly, $X_m \subseteq K_m$. We now prove that $K_m \subseteq X_m$. Note that K_m can be written in the following form:

$$K_m = \big\{y \in \mathbb{R}^m | y_i \geq 0\ \forall i \in [m] \text{ and } \sum_{i=1, i \neq i_0}^m y_i \geq y_{i_0}, \forall i_0 \in [m]\big\}$$
$$= \{y \in \mathbb{R}^m \mid Ay \geq 0\} \text{ where } A \in \mathbb{R}^{2m \times m} \text{ has rank } m$$

By part 2 of Lemma A.4, we then have: $K_m = \text{cone}(R)$ where $R = \bigcup_{\text{extreme rays } R(y) \text{ of } K_m} R(y)$. Therefore, by part 1 of Lemma A.4, it is sufficient to show that if $y \in \mathbb{R}^m$ satisfies any $(m-1)$ equations of K_m with equality, then y should be an element of $\text{cone}\{z \in \{0,1\}^m \mid wt(z) = 2\}$. Note that we have two types of equations:

(I) $\sum_{i=1,\ i \neq i_0}^m y_i - y_{i_0} = 0$ for some $i_0 \in [m]$.
(II) $y_i = 0$ for some $i \in [m]$.

Consider any $(m-1)$ equations of K_m, satisfied with equality. We distinguish two cases:

Case 1: At least $(m-2)$ of those equations are of Type (II). Without loss of generality, we can assume that $y_i = 0$ for all $i \in \{3, \ldots, m\}$. Moreover, since $y \in K_m$, we have that $y_1 - y_2 \geq 0$ and $y_2 - y_1 \geq 0$, which implies that $y_1 = y_2$. Therefore, we conclude that $y = y_1(1\ 1\ 0\ \ldots\ 0)^T \in X_m$.

Case 2: At most $(m-3)$ equations are of Type (II). Hence, at least 2 equations are of Type (I). Without loss of generality, we can assume that $\sum_{i=1,\ i \neq 1}^m y_i = y_1$ and $\sum_{i=1,\ i \neq 2}^m y_i = y_2$. Adding up the last 2 equations, we get $\sum_{i=3}^m y_i = 0$. Since $y \in K_m$, we have $y_i \geq 0$ for all $i \in \{3, \ldots, m\}$. Therefore, we get $y_i = 0$ for all $i \in \{3, \ldots, m\}$. Similarly to Case 1 above, this implies that $y \in X_m$. □

Proof of Theorem 3.2: The "fundamental polytope" P considered by the LP decoder was introduced by [19] and is defined by $P = \bigcap_{j \in C} \text{conv}(C_j)$ where $C_j = \{z \in \{0,1\}^n : wt(z|_{N(j)}) \text{ is even}\}$ for any $j \in C$. For any error pattern $\eta \in \{0,1\}^n$, let $\widetilde{\eta} \in \{-1,1\}^n$ be given by $\widetilde{\eta}_i = (-1)^{\eta_i}$ for all $i \in [n]$. Also, for any $x, y \in \mathbb{R}^n$, let their inner product be $\langle x, y \rangle = \sum_{i=1}^n x_i y_i$. Then, under the all-zeros codeword assumption, there is LP decoding success for η on ζ if and only if the zero vector is the unique optimal solution to the LP in (2), i.e., if and only if $\langle \widetilde{\eta}, 0 \rangle < \langle \widetilde{\eta}, y \rangle$ for every non-zero $y \in P$, which is equivalent to $\widetilde{\eta} \in \text{int}(P^*) = \text{int}(\mathcal{K}^*)$ where $\mathcal{K} = \text{cone}\{P\}$ is the "fundamental cone" and for any $S \subseteq \mathbb{R}^n$, the dual S^* of S is given by $S^* = \{z \in \mathbb{R}^n \mid \langle z, x \rangle \geq 0\ \forall x \in S\}$. By Lemmas A.3 and A.5, we have

$$\begin{aligned}
\mathcal{K} &= \text{cone}\big(\bigcap_{j \in C} \text{conv}(C_j)\big) \\
&= \bigcap_{j \in C} \text{cone}(C_j) \\
&= \bigcap_{j \in C} \text{cone}\{z \in \{0,1\}^n \mid wt(z|_{N(j)}) \text{ is even}\} \\
&= \bigcap_{j \in C} \text{cone}\{z \in \{0,1\}^n \mid wt(z|_{N(j)}) = 2\} \\
&= \bigcap_{j \in C} \big\{y \in (\mathbb{R}_{\geq 0})^n \mid \sum_{i \in N(j) \setminus \{i_0\}} y_i \geq y_{i_0}, \forall i_0 \in N(j)\big\} \\
&= \big\{y \in (\mathbb{R}_{\geq 0})^n \mid \langle y, v_{i_0,j} \rangle \geq 0\ \forall i_0 \in N(j),\ \forall j \in C\big\}
\end{aligned}$$

where $v_{i_0,j} \in \{-1,0,1\}^n$ is defined as follows: For all $i \in [n]$,

$$\big(v_{i_0,j}\big)_i = \begin{cases} 0 & \text{if } i \notin N(j). \\ -1 & \text{if } i = i_0. \\ 1 & \text{if } i \in N(j) \setminus \{i_0\}. \end{cases}$$

Thus,

$$\begin{aligned}
\mathcal{K} &= (\mathbb{R}_{\geq 0})^n \cap \bigcap_{j \in C} \big(\text{cone}\{v_{i_0,j} | i_0 \in N(j)\}\big)^* \\
&= (\mathbb{R}_{\geq 0})^n \cap \bigcap_{j \in C} \big(D_j\big)^*
\end{aligned}$$

where for any $j \in C$, $D_j = \text{cone}\{v_{i_0,j} | i_0 \in N(j)\}$. Note that if $L \subseteq \mathbb{R}^n$ is a cone, then its dual L^* is also a cone. We will use below the following basic properties of dual cones:

i) If $L_1, L_2 \subseteq \mathbb{R}^n$ are cones, then $(L_1 + L_2)^* = L_1^* \cap L_2^*$.
ii) If $L \subseteq \mathbb{R}^n$ is a cone, then $(L^*)^* = \text{cl}(L)$.

[20]Here, $wt(z)$ denotes the Hamming weight of $z \in \{0,1\}^n$, i.e., the number of non-zero coordinates of z.

Therefore, there is LP decoding success for η on $\mathcal{K}$ if and only if $\widetilde{\eta} \in D$ where:

$$\begin{aligned} D &= \mathrm{int}(\mathcal{K}^*) \\ &= \mathrm{int}\left(\left((\mathbb{R}_{\geq 0})^n \cap \bigcap_{j\in C} D_j^*\right)^*\right) \\ &= \mathrm{int}\left(\left(((\mathbb{R}_{\geq 0})^n)^* \cap \bigcap_{j\in C} D_j^*\right)^*\right) \\ &= \mathrm{int}\left(\left(((\mathbb{R}_{\geq 0})^n + \sum_{j\in C} D_j)^*\right)^*\right) \end{aligned}$$

and where the third equality follows from the fact that $(\mathbb{R}_{\geq 0})^n$ is a self-dual cone and the last equality follows from property (i) above. Note that for any $j \in C$, D_j is a cone. Moreover, since $(\mathbb{R}_{\geq 0})^n$ is a cone and the sum of any two cones is also a cone, it follows that $(\mathbb{R}_{\geq 0})^n + \sum_{j\in C} D_j$ is also a cone. Furthermore, by property (ii) above, we get that $D = \mathrm{int}\left(\mathrm{cl}\left((\mathbb{R}_{\geq 0})^n + \sum_{j\in C} D_j\right)\right)$. Being a cone, $(\mathbb{R}_{\geq 0})^n + \sum_{j\in C} D_j$ is a convex set. For any convex set $S \subseteq \mathbb{R}^n$, we have that $\mathrm{int}(\mathrm{cl}(S)) = \mathrm{int}(S)$ (See Lemma 5.28 of [1]). Therefore,

$$\begin{aligned} D &= \mathrm{int}\left((\mathbb{R}_{\geq 0})^n + \sum_{j\in C} D_j\right) \\ &\overset{(a)}{=} (\mathbb{R}_{>0})^n + \sum_{j\in C} D_j \\ &= \Big\{ \sum_{\substack{i_0\in N(j),\\ j\in C}} \lambda_{i_0,j} v_{i_0,j} + u \,|\, \lambda_{i_0,j} \geq 0\ \forall (i_0,j) \text{ and } u \in (\mathbb{R}_{>0})^n \Big\} \end{aligned}$$

where (a) follows from Lemma A.2 and the fact that $\sum_{j\in C} D_j$ is a convex subset of $\mathbb{R}^n$. Thus, there is LP decoding success for η on ζ if and only if there exist $\lambda_{i_0,j} \geq 0$ for all $i_0 \in N(j)$ and all $j \in C$ s.t. $\sum_{i_0\in N(j), j\in C} \lambda_{i_0,j} v_{i_0,j} < \widetilde{\eta}$. Let $wt(i,j) = \left(\sum_{i_0\in N(j)} \lambda_{i_0,j} v_{i_0,j}\right)_i$ for all $i \in [n]$ and all $j \in C$. Since $(v_{i_0,j})_i = 0$ whenever $i \notin N(j)$, we have that for every $i \in [n]$:

$$\begin{aligned} \sum_{j\in N(i)} wt(i,j) &= \sum_{j\in N(i)} \Big(\sum_{i_0\in N(j)} \lambda_{i_0,j} v_{i_0,j} \Big)_i \\ &= \sum_{j\in C} \Big(\sum_{i_0\in N(j)} \lambda_{i_0,j} v_{i_0,j} \Big)_i \\ &= \Big(\sum_{i_0\in N(j), j\in C} \lambda_{i_0,j} v_{i_0,j} \Big)_i < \widetilde{\eta}_i \end{aligned}$$

Moreover, for all $j \in C$, $i_1, i_2 \in N(j)$ s.t. $i_1 \neq i_2$, we have

$$wt(i_1,j) + wt(i_2,j) = \sum_{i_0\in N(j)} \lambda_{i_0,j} \left(\left(v_{i_0,j}\right)_{i_1} + \left(v_{i_0,j}\right)_{i_2} \right) \geq 0$$

since $\left(v_{i_0,j}\right)_{i_1} + \left(v_{i_0,j}\right)_{i_2} \geq 0$ because $i_1 \neq i_2 \in N(j)$. We conclude that LP decoding success for η on ζ is equivalent to the existence of a dual witness for η on ζ. □

B. Analysis of Algorithm 2 and Proof of Theorem 4.1

In this section, we prove the correctness of Algorithm 2 and conclude the proof of Theorem 4.1. We start by stating and proving an algorithm loop invariant that constitutes the main part of the proof of Theorem 4.1. First, we introduce some notation related to the operation of Algorithm 2.

Notation A.6: In the following, let $V = \{v_1, \ldots, v_n\}$. For every $i, j \in [n]$, let $r_{i,j}$ be the number of replicates of variable node v_j after the ith iteration of the algorithm. Moreover, for every $k \in [r_{i,j}]$, let $v_{i,j,k}$ be the kth replicate of v_j after the ith iteration of the algorithm. For all $i \in [n]$, let V_i, C_i, E_i, γ_i and w_i be the set of all variable nodes, set of all check nodes, set of all edges, log-likelihood ratio function and weight function, respectively, after the ith iteration of the algorithm and let $G_i = (V_i, C_i, E_i, w_i, \gamma_i)$. Finally, we set $G_0 = (V_0, C_0, E_0, \gamma_0, w_0)$ to (V, C, E, γ, w).

Lemma A.7: For any $i \geq 0$, after the ith iteration of Algorithm 2, we have[21]

- (I) *For all $j \in [n]$, $\sum_{k=1}^{r_{i,j}} \gamma_i(v_{i,j,k}) = \gamma(v_j)$.*
- (II) *For all $j \in [n]$ and all $k \in [r_{i,j}]$, $\gamma_i(v_{i,j,k})$ has the same sign as $\gamma(v_j)$.*
- (III) *For all $v \in V_i$,*

$$\sum_{c\in N(v): w_i(v,c)>0} w_i(v,c) < \sum_{c\in N(v): w_i(v,c)\leq 0} (-w_i(v,c)) + \gamma_i(v)$$

- (IV) *For all $c \in C_i$, there exist $P_c \geq 0$ and $v \in N(c)$ s.t. $w_i(v,c) = -P_c$ and for all $v' \in N(c)$ s.t. $v' \neq v$, $w_i(v',c) = P_c$.*
- (V) *The directed paths of G are in a bijective correspondence with the directed paths of G_i. Moreover, if the directed path h' of G_i corresponds to the directed path h of G, then the variable and check nodes of h' are replicates of the corresponding variable and check nodes of h.*

Proof of Lemma A.7 (Base Case): Before the first iteration, we have that $r_{0,j} = 1$ and $\gamma_0(v_{0,j,1}) = \gamma(v_j)$ for all $j \in [n]$. Thus, (I) and (II) are initially true. (III) and (IV) are initially true because the original WDAG G satisfies the hyperflow equations (4) and (6). Moreover, (V) is initially true since $G_0 = G$.

Inductive Step: We show that, for every $i \geq 1$, if (I), (III), (IV) and (V) are true after iteration $i-1$ of Algorithm 2, then they are also true after iteration i.

Let $i \geq 1$. In iteration i, a variable node v with log-likelihood ratio $\gamma_{i-1}(v)$ is (possibly) replaced by a number p of replicates $\{v'_1, \ldots, v'_p\}$ with log-likelihood ratios $\left\{\frac{e_l}{e_T^{(v)}}\gamma_{i-1}(v) \mid l \in [p]\right\}$. Therefore, the total sum of the added replicates is $\sum_{l=1}^{p}\left(\frac{e_l}{e_T^{(v)}}\gamma_{i-1}(v)\right) = \gamma_{i-1}(v)$. Thus, (I) is true. By the induction assumption and since $e_l/e_T^{(v)} > 0$, it follows that (II) is also true.

To show that (III) is true, we first note that if $v' \in V_i$ was not created during the ith iteration, then v' will satisfy (III) after the ith iteration. If v' was created during the ith iteration, we distinguish two cases:

In the first case, v' is not a replicate of v (which is the variable node considered in the ith iteration). Then, v' is a replicate of $v_{i-1} \in V_{i-1}$. By the induction assumption, $\gamma_{i-1}(v_{i-1})$ and the weights of the adjacent edges to v_{i-1} satisfy (III) before the ith iteration. Since $\gamma_i(v')$ and the

[21] By "after the 0th iteration", we mean "before the 1st iteration".

weights of the edges adjacent to v' will be respectively equal to $\gamma_{i-1}(v_{i-1})$ and the weights of the edges adjacent to v_{i-1}, scaled by the same positive factor, v' will satisfy (III) after the ith iteration.

In the second case, v' is a replicate of v. Assume that v' is the replicate of v corresponding to the edge (v, c_0) where $c_0 \in N(v)$ and $w_{i-1}(v, c_0) > 0$. During the ith iteration, the subtree corresponding to v' will be created and in this subtree, $\gamma_i(v')$ and the weights of the edges incoming to v' will be respectively equal to $\gamma_{i-1}(v)$ and the weights of the edges incoming to v, scaled by $\theta(v, c_0) = w_{i-1}(v, c_0)/e_T^{(v)}$ where $e_T^{(v)} = \sum_{c \in N(v): w_{i-1}(v,c)>0} w_{i-1}(v, c)$. The only outgoing edge of v' will be (v', c_0). Thus,

$$\begin{aligned}
\sum_{\substack{c \in N(v'):\\ w_i(v',c)>0}} w_i(v', c) &= w_i(v', c_0) = w_{i-1}(v, c_0) \\
&= \theta(v, c_0) \sum_{c \in N(v): w_{i-1}(v,c)>0} w_{i-1}(v, c) \\
&< \theta(v, c_0)\Big(\sum_{\substack{c \in N(v):\\ w_{i-1}(v,c)\le 0}} (-w_{i-1}(v, c)) + \gamma_{i-1}(v)\Big) \\
&= \theta(v, c_0) \sum_{\substack{c \in N(v):\\ w_{i-1}(v,c)\le 0}} (-w_{i-1}(v, c)) + \theta(v, c_0)\gamma_{i-1}(v) \\
&= \sum_{c \in N(v'): w_i(v',c)\le 0} (-w_i(v', c)) + \gamma_i(v')
\end{aligned}$$

Therefore, v' will satisfy (III) after the ith iteration.

Equation (IV) follows from the induction assumption and from the fact that we are either uniformly scaling the neighborhood of a check node or leaving it unchanged.

To prove that (V) is true after the ith iteration, let v be the variable node under consideration in the ith iteration and consider the function that maps the directed path h of G_{i-1} to the directed path h' of G_i as follows:

1) If h does not contain v, then h' is set to h.
2) If h contains v, then h can be uniquely decomposed into the concatenation $h_1 h_2$ where h_1 is a directed path of G_{i-1} that ends at v and h_2 is a directed path of G_{i-1} that starts at v. Let e_l be the first edge of h_2. Then, h' is set to $h'_1 h_2$ where h'_1 is the directed path in the lth created subtree of G' that corresponds to h_1.

This map is a bijection from the set of all directed paths of G_{i-1} to the set of all directed paths of G_i. Moreover, if the directed path h of G_{i-1} is mapped to the directed path h' of G_i, then the variable and check nodes of h' are replicates of the corresponding variable and check nodes of h. □

Proof of Theorem 4.1: Note that 1 and 2 in Theorem 4.1 follow from the operation of Algorithm 2. Moreover, 3, 4, 5 and 6 follow from Lemma A.7 with $\gamma' = \gamma_n$. To prove 7, note that if G has a single sink node v, then v will be the last vertex in any topological ordering of the vertices of G. Furthermore, if v has a single incoming edge with weight α, then it will have only one replicate in T, with a single incoming edge having the same weight α. □

C. Proof of Lemmas 5.10 and 6.9

The goal of this section is prove the following theorem which is used in the proofs of Lemmas 5.10 and 6.9.

Theorem A.8: Let λ, β, m be positive integers with $\beta > d_c - 1$ and $m \ge \lambda$. Consider the optimization problem:

$$v^* = \max_{\substack{(T_0,\ldots,T_h)\in W_h\\ h\in\mathbb{N}, h\ge 1}} f(T_0, \ldots, T_h) \tag{24}$$

where:

$$f(T_0, \ldots, T_h) = \sum_{i=0}^{h} \frac{T_i}{(d_c - 1)^i}$$

and W_h is the set of all tuples $(T_0, \ldots, T_h) \in \mathbb{N}^{h+1}$ satisfying the following three equations:

$$\sum_{i=0}^{h} T_i = m \tag{25}$$

$$T_0 \le \lambda \tag{26}$$

$$T_{i+1} \le \beta T_i \quad \text{for all } i \in \{0, \ldots, h-1\} \tag{27}$$

Then,

$$v^* \le \lambda \frac{\left(\frac{\beta}{d_c-1}\right)^2}{\frac{\beta}{d_c-1} - 1} m^{\frac{\ln\beta - \ln(d_c-1)}{\ln\beta}}$$

We will first prove some lemmas which will lead to Theorem A.8.

Definition A.9: Let $l = \lfloor \log_\beta(\frac{m(\beta-1)}{\lambda} + 1) \rfloor - 1$. Note that $l \ge 0$ since $m \ge \lambda$.

Lemma A.10: Let $(T_0, \ldots, T_h) \in W_h$. Then, $T_i \le \lambda\beta^i$ for all $i \in \{0, \ldots, h\}$.

Proof of Lemma A.10: Follows from equations (26) and (27). □

Lemma A.11: Let

$$T'_i = \lambda\beta^i \textit{ for all } i \in \{0, \ldots, l\}$$

$$T'_{l+1} = m - \lambda\frac{(\beta^{l+1} - 1)}{(\beta - 1)}$$

Then, $(T'_0, \ldots, T'_{l+1}) \in W_{l+1}$.

Proof of Lemma A.11: First, note that $(T'_0, \ldots, T'_{l+1}) \in \mathbb{N}^{l+2}$ since $T'_{l+1} \ge 0$ by Definition A.9. Moreover,

$$\sum_{i=0}^{l+1} T'_i = \sum_{i=0}^{l} \lambda\beta^i + T'_{l+1} = \lambda\frac{(\beta^{l+1} - 1)}{(\beta - 1)} + T'_{l+1} = m$$

We have that $T'_0 \le \lambda$ and for every $i \in \{0, \ldots, l-1\}$, $T'_{i+1} \le \beta T'_i$. We still need to show that $T'_{l+1} \le \beta T'_l$. We proceed by contradiction. Assume that $T'_{l+1} > \beta T'_l$. Then, $T'_{l+1} > \lambda\beta^{l+1}$. Thus,

$$\begin{aligned}
m &= \sum_{i=0}^{l+1} T'_i > \sum_{i=0}^{l+1} \lambda\beta^i = \lambda\frac{(\beta^{l+2} - 1)}{(\beta - 1)} \\
&> \lambda\frac{(\frac{m(\beta-1)}{\lambda} + 1) - 1}{(\beta - 1)} = m
\end{aligned}$$

since $l + 2 = \lfloor \log_\beta(\frac{m(\beta-1)}{\lambda} + 1) \rfloor + 1 > \log_\beta(\frac{m(\beta-1)}{\lambda} + 1)$. □

Lemma A.12: $(T'_0, \ldots, T'_{l+1})$ is the unique (up to leading zeros) element that achieves the maximum in Equation (24).

Proof of Lemma A.12: By Lemma A.11, $(T'_0, \dots, T'_{l+1}) \in W_{l+1}$. Let $(T_0, \dots, T_h) \in W_h$ such that $(T_0, \dots, T_h)$ and $(T'_0, \dots, T'_h)$ are not equal up to leading zeros and without loss of generality assume that $h \geq l+1$ by extending T with zeros if needed. In order to show that $f(T_0, \dots, T_h) < f(T'_0, \dots, T'_h)$, we distinguish two cases:

Case 1: $(T_0, \dots, T_l) \neq (T'_0, \dots, T'_l)$. By Lemma A.10, there exists $k_1 \in \{0, \dots, l\}$ such that $T_{k_1} < \lambda\beta^{k_1}$. Therefore, $\sum_{i=0}^{l} T'_i - \sum_{i=0}^{l} T_i > 0$. Note that:

$$\begin{aligned}
&f(T_0, \dots, T_h) - f(T'_0, \dots, T'_{l+1}) \\
&\quad = \sum_{i=0}^{l} \frac{T_i - T'_i}{(d_c - 1)^i} + \frac{T_{l+1} - T'_{l+1}}{(d_c - 1)^{l+1}} + \sum_{i=l+2}^{h} \frac{T_i}{(d_c - 1)^i} \\
&\quad \leq \frac{1}{(d_c - 1)^l} \sum_{i=0}^{l} (T_i - T'_i) + \frac{T_{l+1} - T'_{l+1}}{(d_c - 1)^{l+1}} \\
&\qquad + \frac{1}{(d_c - 1)^{l+1}} \sum_{i=l+2}^{h} T_i \\
&\quad = \frac{1}{(d_c - 1)^l} \sum_{i=0}^{l} (T_i - T'_i) + \frac{1}{(d_c - 1)^{l+1}} (\sum_{i=l+1}^{h} T_i - T'_{l+1}) \\
&\quad = \frac{1}{(d_c - 1)^l} \sum_{i=0}^{l} (T_i - T'_i) + \frac{1}{(d_c - 1)^{l+1}} \sum_{i=0}^{l} (T'_i - T_i)
\end{aligned}$$

Consequently,

$$\begin{aligned}
f(T_0, \dots, T_h) &\leq f(T'_0, \dots, T'_{l+1}) - \frac{(\sum_{i=0}^{l} T'_i - \sum_{i=0}^{l} T_i)}{(d_c - 1)^l} \\
&\quad + \frac{(\sum_{i=0}^{l} T'_i - \sum_{i=0}^{l} T_i)}{(d_c - 1)^{l+1}} \\
&= f(T'_0, \dots, T'_{l+1}) - (d_c - 2) \frac{(\sum_{i=0}^{l} T'_i - \sum_{i=0}^{l} T_i)}{(d_c - 1)^{l+1}} \\
&< f(T'_0, \dots, T'_{l+1})
\end{aligned}$$

Case 2: $(T_0, \dots, T_l) = (T'_0, \dots, T'_l)$. Then, $T_{l+1} \neq T'_{l+1}$. Since $T'_{l+1} = \sum_{i=l+1}^{h} T_i$, we should have $T'_{l+1} - T_{l+1} > 0$. We have that

$$\begin{aligned}
&f(T_0, \dots, T_h) - f(T'_0, \dots, T'_{l+1}) \\
&\quad = \frac{T_{l+1} - T'_{l+1}}{(d_c - 1)^{l+1}} + \sum_{i=l+2}^{h} \frac{T_i}{(d_c - 1)^i} \\
&\quad \leq \frac{T_{l+1} - T'_{l+1}}{(d_c - 1)^{l+1}} + \frac{1}{(d_c - 1)^{l+2}} \sum_{i=l+2}^{h} T_i \\
&\quad = \frac{T_{l+1} - T'_{l+1}}{(d_c - 1)^{l+1}} + \frac{1}{(d_c - 1)^{l+2}} \sum_{i=0}^{l+1} (T'_i - T_i) \\
&\quad \leq \frac{T_{l+1} - T'_{l+1}}{(d_c - 1)^{l+1}} + \frac{(T'_{l+1} - T_{l+1})}{(d_c - 1)^{l+2}}
\end{aligned}$$

Consequently,

$$\begin{aligned}
f(T_0, \dots, T_h) &\leq f(T'_0, \dots, T'_{l+1}) - \frac{(T'_{l+1} - T_{l+1})}{(d_c - 1)^{l+1}} \\
&\quad + \frac{(T'_{l+1} - T_{l+1})}{(d_c - 1)^{l+2}} \\
&= f(T'_0, \dots, T'_{l+1}) - (d_c - 2) \frac{(T'_{l+1} - T_{l+1})}{(d_c - 1)^{l+2}} \\
&< f(T'_0, \dots, T'_{l+1})
\end{aligned}$$

□

Proof of Theorem A.8: Let $\nu = \beta/(d_c - 1)$. By Lemmas A.12 and A.10, we have that

$$\begin{aligned}
v^* &\leq \sum_{i=0}^{l+1} \frac{T'_i}{(d_c - 1)^i} \\
&\leq \sum_{i=0}^{l+1} \lambda \frac{\beta^i}{(d_c - 1)^i} \\
&= \lambda \sum_{i=0}^{l+1} \nu^i \\
&= \lambda \frac{\nu^{l+2} - 1}{\nu - 1} \\
&< \lambda \frac{\nu^{l+2}}{\nu - 1} \\
&\leq \lambda \frac{\nu^{\log_\beta(\frac{m(\beta-1)}{\lambda}+1)+1}}{\nu - 1} \\
&\leq \lambda \frac{\nu^2}{\nu - 1} \nu^{\log_\beta m} \\
&\leq \lambda \frac{\nu^2}{\nu - 1} m^{\frac{\ln \nu}{\ln \beta}}
\end{aligned}$$

□

D. Proof of Theorem 5.11

The goal of this section is to prove Theorem 5.11 which is restated below.

Theorem 5.11 (Asymptotic Tightness of Theorem 5.1 for (d_v, d_c)-Regular LDPC Codes): There exists an infinite family of (d_v, d_c)-regular Tanner graphs $\{(V_n, C_n, E_n)\}_n$, an infinite family of error patterns $\{\gamma_n\}_n$ and a positive constant c s.t.:

1) *There exists a hyperflow for γ_n on (V_n, C_n, E_n). (Hence, by Theorem 3.7, there exists a WDAG corresponding to a hyperflow for γ_n on (V_n, C_n, E_n).)*
2) *Any WDAG $(V_n, C_n, E_n, w, \gamma_n)$ corresponding to a hyperflow for γ_n on (V_n, C_n, E_n) must have*

$$\max_{e \in E_n} |w(e)| \geq c n^{\frac{\ln(d_v - 1)}{\ln(d_v - 1) + \ln(d_c - 1)}}$$

We now prove some lemmas that lead to the proof of Theorem 5.11.

Definition A.13 (Construction of $\{(V_n, C_n, E_n)\}_n$): Let $\beta = (d_v - 1)(d_c - 1)$. The Tanner graph $\{(V_n, C_n, E_n)\}_n$ is constructed by connecting copies of the following two basic blocks:

1) *The "A block" A_x with parameter the non-negative integer x. A_x is an undirected complete tree rooted at*

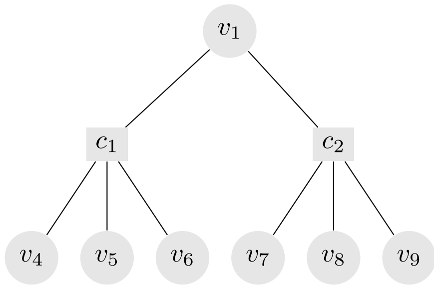

Fig. 8. Example of an A block with parameter $x = 1$ where $d_v = 3$ and $d_c = 4$.

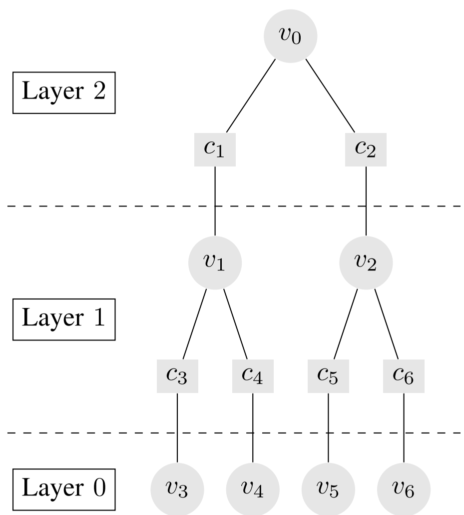

Fig. 9. Example of a B block with parameter $y = 2$ where $d_v = 3$.

a $(d_v - 1)$-regular variable node. The internal nodes of A_x other than the root are either d_c-regular check nodes or d_v-regular variable nodes. The leaves of A_x are all 1-regular variable nodes of depth x.[22] *Thus, A_x has β^x leaves. An example A block is given in Figure 8.*

2) *The "B block" B_y with parameter the non-negative integer y. B_y is an undirected tree rooted at a $(d_v - 1)$-regular variable node. The internal nodes of B_y other than the root are either d_v-regular variable nodes or 2-regular check nodes. The leaves of B_y are 1-regular variable nodes. The nodes of B_y are divided into $y + 1$ layers indexed from y to 0. Layer y consists of the root and the $(d_v - 1)$ check nodes that are connected to the root. Each check node in layer i is connected to a single variable node in layer $i - 1$ for all $i = y, y - 1, \dots, 1$. Each variable node in layer i is connected to $d_v - 1$ check nodes in the same layer for all $i = y, y-1, \dots, 1$. Thus, layer 0 consists of $(d_v - 1)^y$ leaves which are all 1-regular variable nodes. An example B block is given in Figure 9.*

Let $\gamma = \frac{\ln(d_v-1)}{\ln(d_v-1)+\ln(d_c-1)}$. For every non-negative integer n, let $y_n = \lfloor \log_{(d_v-1)} n^\gamma \rfloor$ and $b_n = (d_v-1)^{y_n} = \Theta(n^\gamma)$. The Tanner graph $\{(V_n, C_n, E_n)\}_n$ is constructed using a root check node, one B block, many A blocks and some auxiliary variable and check nodes as follows:

1) *Start with a check node c_0.*
2) *Connect c_0 to the roots of $d_c - 1$ A_{y_n+1} blocks and to the root of one B_{y_n} block. Note that B_{y_n} has b_n leaves.*
3) *For every $i = y_n, y_{n-1}, \dots, 1$, connect each check node in layer i of B_{y_n} to the roots of $(d_c - 2)$ A_i blocks. Note that there are $(d_v - 1)^{y_n - i + 1}$ check nodes in layer i.*
4) *Let T_n be the tree constructed so far and l_n be its number of leaves. Note that all the leaves of T_n are 1-regular variable nodes. Complete T_n into a (d_v, d_c)-regular graph by adding $O(l_n)$ d_c-regular new check nodes and (if needed) $O(l_n)$ d_v-regular new variable nodes in such a way that each new check is either connected to zero or to at least two leaves of the B block.*[23]

We call the check and variable nodes added in step 4 the "connecting" check and variable nodes respectively.

Definition A.14 (Construction of $\{\gamma_n\}_n$): Let $\{(V_n, C_n, E_n)\}_n$ be the Tanner graph given in Definition A.13. The error pattern γ_n is defined by:

1) *For every variable node v in an A block, $\gamma_n(v) = 1$.*
2) *For every variable node v in the B block, $\gamma_n(v) = -1$.*
3) *For every connecting variable node v, $\gamma_n(v) = 1$.*

Lemma A.15 (Size of the Code): For any positive integer n, the Tanner graph $\{(V_n, C_n, E_n)\}_n$ given in Definition A.13 is a (d_v, d_c)-regular code with $\Theta(n)$ variable nodes.

Proof of Lemma A.15: It is enough to show that the number l_n of leaves of T_n is $O(n)$. The number of leaves of block B_{y_n} is $b_n = \Theta(n^\gamma)$. The number of leaves of block A_y is $(d_v-1)^y$. Thus, the number of leaves in all the A-blocks is

$$\begin{aligned} a_n &= (d_c - 1)(d_v - 1)^{y_n+1} + (d_c - 2)\sum_{i=1}^{y_n}(d_v - 1)^{y_n-i+1}\beta^i \\ &= O((d_v - 1)^{y_n}) + O((d_v - 1)^{y_n}\sum_{i=1}^{y_n}(d_c - 1)^i) \\ &= O(b_n + \beta^{y_n}) \end{aligned}$$

because $(d_v - 1)^{y_n} = b_n$ and $\sum_{i=1}^{y_n}(d_v - 1)^i = O((d_c - 1)^{y_n})$. Since $\beta^{y_n} = \Theta(n)$ and $b_n = o(n)$, we get that $l_n = b_n + a_n = \Theta(n)$. □

Lemma A.16 (Existence of a Hyperflow for $\{\gamma_n\}_n$ on $\{(V_n, C_n, E_n)\}_n$): Let $\{(V_n, C_n, E_n)\}_n$ be the Tanner graph given in Definition A.13 and let γ_n be the error pattern given in Definition A.14. Then, for every positive integer n, there exists a hyperflow for γ_n on (V_n, C_n, E_n).

Proof of Lemma A.16: Let $\epsilon > 0$. We will further specify ϵ at the end of the proof. Consider the following assignment of weigths to edges of E_n:

1) In every A block, the edges are directed toward the root of the block. The edges outgoing from the leaves have weight $1 - \epsilon$. For every check node, the weight of the outgoing edge is equal to the common weight of its incoming edges. For each variable node, the sum of the weights of the outgoing edges is equal to the sum

[22]The depth of a variable node v is the number of check nodes on the unique path from the root to v.

[23]Note that if $(d_v - 1)l_n$ is divisible by d_c, we don't need any extra variable nodes. In the worst case, we can add d_c copies of T_n so that $(d_v - 1)d_c l_n$ is divisible by d_c.

of the weights of the incoming edges plus $1-\epsilon$. Thus, the weight of the edge outgoing from the root of the A_x block is

$$r_x = (1-\epsilon)\sum_{t=0}^{x}(d_\text{v}-1)^t = (1-\epsilon)\frac{(d_\text{v}-1)^{x+1}-1}{d_\text{v}-2}$$

2) In the B block, the edges are directed toward the leaves. The edge connecting c_0 to the root of block B has weight w_{y_n} where for any $i \in \{0,\ldots,y_n\}$:

$$w_i := (1+\epsilon)\sum_{j=0}^{i}(d_\text{v}-1)^j = (1+\epsilon)\frac{(d_\text{v}-1)^{i+1}-1}{d_\text{v}-2}$$

For every internal variable node v, the weight of each outgoing edge from v is $\frac{z-(1+\epsilon)}{d_\text{v}-1}$ where z is the weight of the edge incoming to v. For every internal check node c, the weight of the edge outgoing from c is equal to the weight of the edge incoming to c. By induction on the layer index $i = y_n, y_{n-1},\ldots,0$, for every variable node v in layer i, the weight of its incoming edge is w_i and (if v is not a leaf) the weight of each of its outgoing edges is w_{i-1} (since w_i satisfies the recurrence $w_{i-1} = \frac{w_i-(1+\epsilon)}{d_\text{v}-1}$ for all $i = y_n, y_{n-1},\ldots,1$).

3) All edges adjacent to connecting check or variable nodes have weight zero.

By construction, the weights satisfy the dual witness equations (4) and (5) for all check and variable nodes in A blocks, all internal variable nodes in the B block and all the connecting check and variable nodes. To guarantee that equations (4) and (5) hold for the root check node c_0, we need that $r_{y_{n+1}} \geq w_{y_n}$. To guarantee them for the internal check nodes of the B block, we need that $r_{i+1} \geq w_i$ for all $i = y_n-1,\ldots,1$. To guarantee them for the leaves of the B block, we need that $w_0 - 1 > 0$, which holds since $w_0 = 1+\epsilon$. Thus, for every $i = y_n, y_{n-1},\ldots,1$, we need that $r_{i+1} \geq w_i$, i.e.,

$$(1-\epsilon)\frac{(d_\text{v}-1)^{i+2}-1}{d_\text{v}-2} \geq (1+\epsilon)\frac{(d_\text{v}-1)^{i+1}-1}{d_\text{v}-2}$$

which can be guaranteed by letting $0 < \epsilon < 1-\frac{2}{d_\text{v}}$. □

Lemma A.17 (Lower Bound for Any Hyperflow for $\{\gamma_n\}_n$ on $\{(V_n,C_n,E_n)\}_n$): For any positive integer n, any WDAG (V_n,C_n,E_n,w,γ_n) corresponding to a hyperflow for γ_n on (V_n,C_n,E_n) must have

$$\max_{e\in E_n}|w(e)| \geq cn^{\frac{\ln(d_\text{v}-1)}{\ln(d_\text{v}-1)+\ln(d_\text{c}-1)}}$$

for some constant $c > 0$.

Proof of Lemma A.17: Let (V_n,C_n,E_n,w,γ_n) be a WDAG corresponding to a hyperflow for γ_n on (V_n,C_n,E_n). Since $\gamma_n(v) = -1$ for every leaf v of the B block (which has b_n leaves) and since each connecting check node adjacent to a leaf of the B block is connected to at least two leaves of the B block, there should be a flow of total value larger than b_n from the non-leaf and non-connecting nodes of the B block to its leaves. Applying the same argument inductively and using the fact that for every variable node v of the B block $\gamma_n(v) = -1$, we get that all the edges of the B block should be oriented toward its leaves and that there should be a flow of value larger than b_n entering the root of the B block. Thus, the edge connecting c_0 to the root of the B block should be oriented toward the B block and should have value larger than $b_n = \Theta(n^{\frac{\ln(d_\text{v}-1)}{\ln(d_\text{v}-1)+\ln(d_\text{c}-1)}})$. □

Proof of Theorem 5.11: Follows from Lemmas A.15, A.16 and A.17. □

Acknowledgments

The authors would like to thank Pascal Vontobel and the anonymous reviewers for very helpful comments.

References

[1] C. D. Aliprantis and K. C. Border, *Infinite Dimensional Analysis: A Hitchhiker's Guide*. New York, NY, USA: Springer-Verlag, 2006.
[2] L. Bazzi, B. Ghazi, and R. Urbanke, "Linear programming decoding of spatially coupled codes," in *Proc. IEEE ISIT*, Jul. 2013, pp. 1486–1490.
[3] E. Berlekamp, R. J. McEliece, and H. C. A. V. Tilborg, "On the inherent intractability of certain coding problems (corresp.)," *IEEE Trans. Inf. Theory*, vol. 24, no. 3, pp. 384–386, May 1978.
[4] D. Burshtein, "Iterative approximate linear programming decoding of LDPC codes with linear complexity," *IEEE Trans. Inf. Theory*, vol. 55, no. 11, pp. 4835–4859, Nov. 2009.
[5] D. Burshtein, private communication, 2011.
[6] C. Daskalakis, A. G. Dimakis, R. M. Karp, and M. J. Wainwright, "Probabilistic analysis of linear programming decoding," *IEEE Trans. Inf. Theory*, vol. 54, no. 8, pp. 3565–3578, Aug. 2008.
[7] D. L. Donoho, A. Javanmard, and A. Montanari, "Information-theoretically optimal compressed sensing via spatial coupling and approximate message passing," in *Proc. IEEE ISIT*, Cambridge, MA, USA, Jul. 2012, pp. 1231–1235.
[8] J. Feldman, "Decoding error-correcting codes via linear programming," Ph.D. dissertation, Dept. Elect. Eng. Comput. Sci., Massachusetts Inst. Technol., Cambridge, MA, USA, 2003.
[9] J. Feldman, T. Malkin, R. A. Servedio, C. Stein, and M. J. Wainwright, "LP decoding corrects a constant fraction of errors," *IEEE Trans. Inf. Theory*, vol. 53, no. 1, pp. 82–89, Jan. 2007.
[10] J. Feldman and C. Stein, "LP decoding achieves capacity," in *Proc. 16th Annu. ACM-SIAM SODA*, Philadelphia, PA, USA, 2005, pp. 460–469.
[11] J. Feldman, M. J. Wainwright, and D. R. Karger, "Using linear programming to decode binary linear codes," *IEEE Trans. Inf. Theory*, vol. 51, no. 3, pp. 954–972, Mar. 2005.
[12] R. G. Gallager, "Low-density parity-check codes," *IEEE Trans. Inf. Theory*, vol. 8, no. 1, pp. 21–28, Jan. 1962.
[13] N. Halabi and G. Even, "Linear-programming decoding of Tanner codes with local-optimality certificates," in *Proc. IEEE ISIT*, Cambridge, MA, USA, 2012, pp. 2686–2690.
[14] A. J. Felstrom and K. S. Zigangirov, "Time-varying periodic convolutional codes with low-density parity-check matrix," *IEEE Trans. Inf. Theory*, vol. 45, no. 6, pp. 2181–2191, Sep. 1999.
[15] F. Krzakala, M. Mézard, F. Sausset, Y. F. Sun, and L. Zdeborová, "Statistical-physics-based reconstruction in compressed sensing," *Phys. Rev. X*, vol. 2, no. 2, p. 021005, May 2012.
[16] F. Krzakala, M. Mézard, F. Sausset, Y. Sun, and L. Zdeborová, "Probabilistic reconstruction in compressed sensing: Algorithms, phase diagrams, and threshold achieving matrices," *J. Statist. Mech., Theory Experim.*, vol. 2012, no. 08, p. 08009, Jun. 2012.
[17] S. Kudekar, T. J. Richardson, and R. L. Urbanke, "Threshold Saturation via Spatial Coupling: Why Convolutional LDPC Ensembles Perform So Well over the BEC," *IEEE Trans. Inf. Theory*, vol. 57, no. 2, pp. 803–834, Feb. 2011.
[18] S. Kudekar, T. Richardson, and R. Urbanke, "Spatially coupled ensembles universally achieve capacity under belief propagation," in *Proc. IEEE ISIT*, Cambridge, MA, USA, Jul. 2012, pp. 453–457.
[19] R. Koetter and P. O. Vontobel, "Graph-covers and iterative decoding of finite length codes," in *Proc. IEEE Int. Symp. Turbo Codes Appl.*, 2003, pp. 75–82.
[20] P. M. Olmos and R. Urbanke, "Scaling behavior of convolutional LDPC ensembles over the BEC," in *Proc. IEEE ISIT*, St. Petersburg, Russia, Feb. 2011, pp. 1816–1820.

[21] A. Schrijver, *Theory of Linear and Integer Programming*. New York, NY, USA: Wiley, 1998.

[22] M. B. S. Tavares, K. S. Zigangirov, and G. P. Fettweis, "Tail-biting LDPC convolutional codes," in *Proc. IEEE ISIT*, Nice, France, Jun. 2007, pp. 2341–2345.

[23] P. O. Vontobel and R. Koetter, "Towards low-complexity linear-programming decoding," in *Proc. 6th Int. ITG-Conf. Sour. Channel Coding, 4th Int. Symp. Turbo Codes Rel. Topics*, Apr. 2006, pp. 1–9.

[24] P. O. Vontobel and R. Koetter, "On low-complexity linear-programming decoding of LDPC codes," *Eur. Trans. Telecommun.*, vol. 18, no. 5, pp. 509–517, Aug. 2007.

Louay Bazzi received his Ph.D. degree from the department of Electrical Engineering and Computer Science at MIT in 2003. He is currently an associate Professor in the Electrical and Computer Engineering department at the American University of Beirut. His research interests include coding theory, pseudorandomness, and complexity theory.

Badih Ghazi received his BE in Computer and Communications Engineering at the American University of Beirut in 2012. He is currently a graduate student at the EECS department at MIT. His research interests include coding theory and complexity theory.

Rüdiger L. Urbanke obtained his Dipl. Ing. degree from the Vienna University of Technology, Austria in 1990 and the M.Sc. and PhD degrees in Electrical Engineering from Washington University in St. Louis, MO, in 1992 and 1995, respectively.

He held a position at the Mathematics of Communications Department at Bell Labs from 1995 till 1999 before becoming a faculty member at the School of Computer & Communication Sciences of EPFL. He is a member of the Information Processing Group.

He is principally interested in the analysis and design of iterative coding schemes, which allow reliable transmission close to theoretical limits at low complexities. Such schemes are part of most modern communications standards, including wireless transmission, optical communication and hard disk storage. More broadly, his research focuses on the analysis of graphical models and the application of methods from statistical physics to problems in communications.

From 2000–2004 he was an Associate Editor of the IEEE TRANSACTIONS ON INFORMATION THEORY and he is currently on the board of the series *Foundations and Trends in Communications and Information Theory*. Since 2013 he has been a Member of the Board of the Information Theory Society as well as a Distinguished Speaker. In 2013 he served a short stint as Dean a. i. of I&C.

Dr. Urbanke is a recipient of a Fulbright Scholarship. He is a co-author of the book *Modern Coding Theory* (Cambridge University Press) a co-recipient of the 2002 and the 2013 IEEE Information Theory Society Paper Award, the 2011 IEEE Koji Kobayashi Award, as well as the 2014 IEEE Hamming Medal.