Asymptotic Analysis and Spatial Coupling of Counter Braids

Citation for the original published paper (version of record):
Rosnes, E., Graell i Amat, A. (2018). Asymptotic Analysis and Spatial Coupling of Counter Braids.
IEEE Transactions on Information Theory, 64(11): 7242-7263.
http://dx.doi.org/10.1109/TIT.2018.2864945

Asymptotic Analysis and Spatial Coupling of Counter Braids

Eirik Rosnes, *Senior Member, IEEE* and Alexandre Graell i Amat, *Senior Member, IEEE*

***Abstract*—A counter braid (CB) is a novel counter architecture introduced by Lu *et al.* in 2007 for per-flow measurements on high-speed links which can be decoded with low complexity using message passing (MP). CBs achieve an asymptotic compression rate (under optimal decoding) that matches the entropy lower bound of the flow size distribution. In this paper, we apply the concept of spatial coupling to CBs to improve the performance of the original CBs and analyze the performance of the resulting spatially-coupled CBs (SC-CBs). We introduce an equivalent bipartite graph representation of CBs with identical iteration-by-iteration finite-length and asymptotic performance. Based on this equivalent representation, we then analyze the asymptotic performance of single-layer CBs and SC-CBs under the MP decoding algorithm proposed by Lu *et al.*. In particular, we derive the potential threshold of the uncoupled system and show that it is equal to the area threshold. We also derive the Maxwell decoder for CBs and prove that the potential threshold is an upper bound on the Maxwell decoding threshold, which, in turn, is a lower bound on the maximum *a posteriori* (MAP) decoding threshold. We then show that the area under the extended MP extrinsic information transfer curve (defined for the equivalent graph), computed for the expected residual CB graph when a peeling decoder equivalent to the MP decoder stops, is equal to zero precisely at the area threshold. This, combined with the analysis of the Maxwell decoder and simulation results, leads us to the conjecture that the potential threshold is in fact equal to the Maxwell decoding threshold and hence a lower bound on the MAP decoding threshold. Interestingly, SC-CBs do not show the well-known phenomenon of threshold saturation of the MP decoding threshold to the potential threshold characteristic of spatially-coupled low-density parity-check codes and other coupled systems. However, SC-CBs yield better MP decoding thresholds than their uncoupled counterparts. Finally, we also consider SC-CBs as a compressed sensing scheme and show that low undersampling factors can be achieved.**

***Index Terms*—Compressed sensing, counter braids, density evolution, Maxwell decoder, spatial coupling.**

I. Introduction

Traffic measurement in large scale networks is key to network operators for network management, for example, in terms of pricing, billing, and diagnosing of network problems. Traffic measurement consists of measuring the size of network flows. A network flow is defined as a sequence of packets that traverse the network and that share a number of properties, e.g., the same source and destination. The size of a flow is the number of packets of a particular kind. On high-speed links, the inter-arrival time between packets can be as short as 40 nanoseconds for a 10 Gbps link. Therefore, an exact estimation of the flow sizes is technologically challenging and even expensive to build, since it requires large arrays of high-speed counters. Thus, the design of efficient, low-complexity algorithms to measure the size of active flows in high-speed networks is of great practical interest.

Recently, Lu *et al.* proposed a novel counter architecture, inspired by sparse graph codes, for measuring network flow sizes, nicknamed Counter Braid (CB) [1]–[3]. CBs address the problem of cheap high-speed memory-efficient approximate counting. In particular, they use less memory space than other approximate counting techniques, since the flow sizes are compressed *on-the-fly*. CBs are asymptotically optimal (under some mild conditions) [3], i.e., the average number of bits needed to store the size of a flow tends to the information-theoretic limit (under maximum-likelihood (ML) decoding) when the number of flows goes to infinity. Furthermore, they are characterized by a layered structure which can be described by a graph. In [3], a low-complexity message passing (MP) decoding algorithm working on the resulting graph was proposed, where the messages exchanged within the graph are positive integers. In this sense, no reliability information is exchanged in the iterative process and the algorithm can be assimilated to hard-decision MP decoding algorithms for low-density parity-check (LDPC) codes. In general, good performance can be achieved with a small number of layers.

A single-layer CB can also be regarded as a compressed sensing (CS) scheme for nonnegative (integer) signals [2]. CS establishes that sparse signals can be recovered from significantly fewer samples as compared to conventional Nyquist sampling. Indeed, the decoding of single-layer CBs can also be seen as a sparse signal recovery problem. In particular, if we set to zero the entries of minimum flow size in the vector of flow sizes, then it can be interpreted as a nonnegative t-sparse vector with t nonzero (integer) entries. In this setup, the *counters* of the CB architecture perform linear measurements of the sparse signal, i.e., each counter corresponds to a linear measurement in the CS context. In [2], the authors analyzed the threshold (when the number of flows goes to infinity) on the undersampling factor (the number of counters per flow, corresponding to the ratio between the number of measurements and the length of the sparse vector) of CBs above which recovery of the sparse vector is possible. It was shown that it is very close to the corresponding threshold by Donoho and Tanner [4] on the undersampling factor such

Parts of this paper have been presented at the IEEE Information Theory Workshop, Hobart, Australia, Nov. 2014, and at the IEEE Information Theory Workshop, Jeju Island, Korea, Oct. 2015.

E. Rosnes was supported by the Norwegian-Estonian Research Cooperation Programme (grant EMP133) and by the Research Council of Norway (grant 240985/F20). A. Graell i Amat was supported by the Swedish Research Council under grants #2011-5961 and #2016-04253.

E. Rosnes is with Simula@UiB, N-5020 Bergen, Norway (e-mail: eirikrosnes@simula.no).

A. Graell i Amat is with the Department of Electrical Engineering, Chalmers University of Technology, SE-41296 Gothenburg, Sweden (e-mail: alexandre.graell@chalmers.se).

that most signals of a given sparsity can be recovered exactly using ℓ_1-norm minimization reconstruction, assuming random Gaussian measurement matrices and nonnegative real signals.

Spatial coupling of LDPC codes [5]–[7] has revealed as a powerful technique that improves the belief propagation decoding threshold of the spatially-coupled LDPC (SC-LDPC) code to the maximum *a posteriori* (MAP) decoding threshold of the underlying block code ensemble, a phenomenon known as threshold saturation. The concept of spatial coupling is not exclusive of LDPC codes, and it applies to other classes of codes such as turbo-like codes [8], [9] and to iterative hard-decision decoding (HDD) of spatially-coupled (SC) generalized LDPC codes [10], [11] as well as to other scenarios, such as relaying [12], lossy compression [13], joint iterative decoding of LDPC codes on intersymbol-interference channels with erasure noise where decoding is performed on a large graph representing both the channel and the code constraints [14], code-division multiple-access [15], and CS [16]. In all these cases, threshold saturation to the so-called *potential threshold* has been proven or observed numerically, while threshold saturation to the so-called *condensation threshold* was observed for lossy compression in [17]. In [18], it was observed that the *survey propagation threshold* increases and saturates towards to the *phase transition threshold* for SC constraint satisfaction problems, as well as saturation of the *dynamic threshold* towards the condensation threshold.

In this paper, we apply the concept of spatial coupling to CBs, and show that they yield improved MP decoding thresholds as compared to uncoupled CBs. Furthermore, we analyze the asymptotic performance of single-layer spatially-coupled CBs (SC-CBs). We derive an equivalent bipartite graph representation of CBs, with identical iteration-by-iteration finite-length performance and asymptotic behavior to that of CBs decoded on the original bipartite graph. Based on this equivalent representation, we prove that the potential threshold, introduced by Yedla *et al.* in [19], and the area threshold are equal. We further derive the Maxwell decoder [20] for CBs and prove that the area threshold is an upper bound on the Maxwell decoding threshold, which, in turn, is a lower bound on the MAP decoding threshold. We also show that the area under the extended MP (EMP) extrinsic information transfer (EXIT) curve (defined for the equivalent graph), computed for the expected residual CB graph when a peeling decoder equivalent to the MP decoder stops, is equal to zero precisely at the area threshold. This result, combined with an asymptotic analysis of the Maxwell decoder, leads us to formulate the conjecture that the potential threshold is equal to the Maxwell decoding threshold and hence a lower bound on the MAP decoding threshold. This conjecture is also supported by simulation results.

Interestingly, we observe that SC-CBs do not show threshold saturation of the MP decoding threshold to the potential threshold. Indeed, when coupling the original (or the equivalent) graph, there is a remaining gap between the potential threshold (conjectured to be equal to the Maxwell decoding threshold) and the MP decoding threshold of SC-CBs even in the limit of large coupling chain length and smoothing parameter. The lack of threshold saturation seems to be fundamental and due to the fact that the flow node update rule for even and odd iterations is different. To the best of our knowledge, this is one of the rear coupled systems where threshold saturation does not occur.

We also discuss the construction of multilayer SC-CBs and the extension of the density evolution (DE) analysis to this case. Finally, we discuss single-layer SC-CBs in the context of CS. We compare the threshold on the undersampling factor of SC-CBs with that of CBs [2] and with the threshold in [4], and we show that lower thresholds can be achieved.

The remainder of the paper is organized as follows. In Section II, we briefly review CBs and the corresponding MP decoding algorithm. In Section III, we introduce an equivalent bipartite graph representation of CBs. In Section IV, we give an asymptotic analysis of single-layer CBs, based on the equivalent graph representation of Section III, and show that the area threshold and the potential threshold are equal. In Section V, we derive the Maxwell decoder for CBs and formulate the conjecture that the potential threshold is a lower bound on the MAP decoding threshold. Section VI introduces SC-CBs, and Section VII gives numerical results on their asymptotic and finite-length performance. A discussion on the lack of threshold saturation is provided in Section VIII. Finally, the connection with CS is discussed in Section IX, and Section X draws some conclusions.

II. Counter Braids

A CB is a counter architecture consisting of $L \geq 1$ layers. At layer $l = 1, \ldots, L$, it has m_l counters of depth d_l bits, with $m_i < m_j$ for $i > j$. The number of distinct flows to be counted is denoted by m_0.

The l-th layer of a CB can be represented by a bipartite graph $\mathcal{G}_l = \mathcal{G}_l(\mathcal{F}_l \cup \mathcal{C}_l, \mathcal{E}_l)$, where $\mathcal{F}_l$ (of size m_{l-1}) denotes the set of flow nodes, and $\mathcal{C}_l$ (of size m_l) denotes the set of counter nodes. The set of edges of the graph is denoted by $\mathcal{E}_l$. The l-th layer is connected to the $(l-1)$-th layer by a bijective mapping $\Xi_l(\mathsf{f}) = \mathsf{c}$ on the set of flow nodes of the l-th layer, where $\mathsf{f} \in \mathcal{F}_l$ and $\mathsf{c} \in \mathcal{C}_{l-1}$, $l = 2, \ldots, L$. For notational convenience, the neighborhood of a node a (either a counter node c or a flow node f) is denoted by $\Gamma(\mathsf{a})$.

When a flow is encountered (for instance, on a high-speed link), all connected counter nodes of the first layer are incremented modulo 2^{d_1}. If a counter node c of the first layer overflows, all connected counter nodes of the second layer (formally the counter nodes in the set $\Gamma(\Xi_2^{-1}(\mathsf{c}))$), are incremented modulo 2^{d_2}. Furthermore, if a counter node in the second layer overflows, all connected counter nodes of the third layer are incremented modulo 2^{d_3}. This process is repeated for each level until we reach the final layer L. We denote by $\phi(\mathsf{c})$ the final value of a counter node c prior to decoding, and by $\hat{\phi}(\mathsf{f})$ the estimated value (after decoding) of a flow node f. The corresponding actual flow size is denoted by $\phi(\mathsf{f})$.

An example of a two-layer CB is shown in Fig. 1, where flow nodes are represented by empty circles and counter nodes by filled squares. Fig. $1(a)$ shows the bipartite graphs $\mathcal{G}_1$ and $\mathcal{G}_2$ of the two-layer CB, while Fig. $1(b)$ depicts an

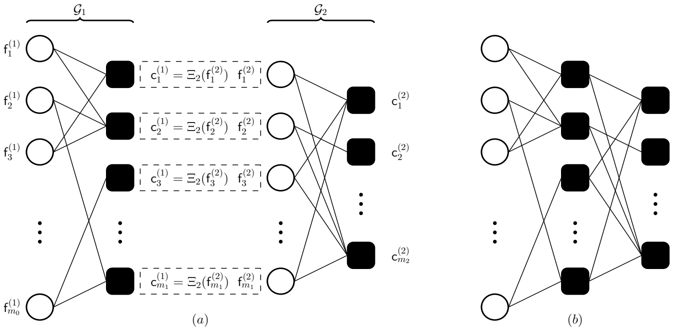
$\mathcal{G}_1$
$\mathcal{G}_2$
$\mathsf{f}_1^{(1)}$
$\mathsf{f}_2^{(1)}$
$\mathsf{f}_3^{(1)}$
$\mathsf{f}_{m_0}^{(1)}$
$\mathsf{c}_1^{(1)} = \Xi_2(\mathsf{f}_1^{(2)})$ $\mathsf{f}_1^{(2)}$
$\mathsf{c}_2^{(1)} = \Xi_2(\mathsf{f}_2^{(2)})$ $\mathsf{f}_2^{(2)}$
$\mathsf{c}_3^{(1)} = \Xi_2(\mathsf{f}_3^{(2)})$ $\mathsf{f}_3^{(2)}$
$\mathsf{c}_{m_1}^{(1)} = \Xi_2(\mathsf{f}_{m_1}^{(2)})$ $\mathsf{f}_{m_1}^{(2)}$
$\mathsf{c}_1^{(2)}$
$\mathsf{c}_2^{(2)}$
$\mathsf{c}_{m_2}^{(2)}$
(a)
(b)

Fig. 1. The graph of a two-layer CB. (a) The bipartite graphs $\mathcal{G}_1$ and $\mathcal{G}_2$; (b) the combined graph.

equivalent graph where a flow node of the second layer and its corresponding counter node of the first layer in Fig. $1(a)$ are represented using a single *combined* counter node due to the bijective mapping $\Xi_2(\cdot)$.

In the following, we will assume that the bipartite graphs $\mathcal{G}_l$, $l = 1, \ldots, L$, are left-regular, i.e., $|\Gamma(\mathsf{f})| = k_l$, for some integer $k_l \geq 2$, and for all $\mathsf{f} \in \mathcal{F}_l$. The assignment of graph sockets of flow nodes to graph sockets of counter nodes, i.e., the connections in the graph, is done in a random fashion. This means that, asymptotically as the number of flow nodes tends to infinity (while $\frac{|\mathcal{C}_l|}{|\mathcal{F}_l|}$ is kept fixed), the distribution of the fraction of edges connected to a given counter node approaches a Poisson distribution [1]. We can now formally define the flow and counter node degree distributions. For counter nodes, we assume the asymptotic Poisson distribution. Let

$$L_l(z) = z^{k_l}, \qquad\qquad k_l \geq 2,$$

and

$$R_l(z) = \sum_{i=0}^{\infty} R_i^{(l)} z^i = \sum_{i=0}^{\infty} \frac{\mathrm{e}^{-\gamma_l}(\gamma_l z)^i}{i!}$$

denote the node-perspective flow node and counter node degree distributions of the l-th layer, respectively. Here, $\gamma_l = \frac{m_{l-1}k_l}{m_l}$ is the average counter node degree. The corresponding edge-perspective degree distributions are

$$\lambda_l(z) = \frac{L_l'(z)}{L_l'(1)} = z^{k_l - 1}$$

and

$$\rho_l(z) = \frac{R_l'(z)}{R_l'(1)} = \sum_{i=0}^{\infty} \frac{\mathrm{e}^{-\gamma_l}(\gamma_l z)^i}{i!} = R_l(z),$$

respectively. Finally, we denote by $\beta_l = \frac{m_l}{m_{l-1}}$ the fraction of the number of counter and flow nodes of the l-th layer.

A. Message Passing Decoder

CBs can be decoded using an MP decoding algorithm [1] on the corresponding graph. In this section, we present the low-complexity MP decoding algorithm proposed in [1]. We consider first the sum-product (SP) algorithm to decode CBs and motivate the derivation of the algorithm in [1]. The derivation was not discussed in [1] (or in [2], [3], [21]), but may help in gaining some insight into the algorithm.

For simplicity, we consider a single-layer CB, and denote its set of edges by $\mathcal{E} = \mathcal{E}_1$. The optimality criterion behind the SP algorithm for CBs is the *symbol-wise* MAP,

$$\hat{\phi}_{\text{MAP}}(\mathsf{f}_i) = \arg\max_{\phi(\mathsf{f})} \Pr(\phi(\mathsf{f}_i) = \phi(\mathsf{f}) | \phi(\mathbf{c})), \tag{1}$$

where $\phi(\mathbf{c}) = (\phi(\mathsf{c}_1), \ldots, \phi(\mathsf{c}_{m_1}))$.

Implementing the MAP decoder to solve (1) directly is computationally infeasible. The SP algorithm attempts to solve (1) approximately operating on the factor graph of the CB. Let $\psi_{\mathsf{c}\to\mathsf{f}}^{\text{SP},(\ell)}(a)$ denote the message sent from counter node c to flow node f at iteration ℓ of the SP algorithm, corresponding to the probability that the flow f is of size a, i.e., $\phi(\mathsf{f}) = a$. Note that a is a nonnegative integer that runs from $f_{\min}$ to $\phi(\mathsf{c}) - (|\Gamma(\mathsf{c})| - 1)f_{\min}$, where $f_{\min}$ is the minimum flow size (from the flow size distribution) of a flow node. Furthermore, let $\mu_{\mathsf{f}\to\mathsf{c}}^{\text{SP},(\ell)}(a)$ denote the message sent from flow node f to counter node c at iteration ℓ, corresponding to the probability that the flow f is of size a. The update rules for $\psi_{\mathsf{c}\to\mathsf{f}}^{\text{SP},(\ell)}(a)$ and $\mu_{\mathsf{f}\to\mathsf{c}}^{\text{SP},(\ell)}(a)$ can be written as [22]

$$\psi_{\mathsf{c}\to\mathsf{f}}^{\text{SP},(\ell)}(a) = \sum_{\substack{(a_{\mathsf{f}'}\,:\,\mathsf{f}'\in\Gamma(\mathsf{c})\backslash\mathsf{f})\,:\,a_{\mathsf{f}'}\geq f_{\min}\\ \text{and}\sum_{\mathsf{f}'\in\Gamma(\mathsf{c})\backslash\mathsf{f}} a_{\mathsf{f}'}=\phi(\mathsf{c})-a}} \prod_{\mathsf{f}'\in\Gamma(\mathsf{c})\backslash\mathsf{f}} \mu_{\mathsf{f}'\to\mathsf{c}}^{\text{SP},(\ell-1)}(a_{\mathsf{f}'}) \tag{2}$$

and

$$\mu_{\mathsf{f}\to\mathsf{c}}^{\text{SP},(\ell)}(a) = \prod_{\mathsf{c}'\in\Gamma(\mathsf{f})\backslash\mathsf{c}} \psi_{\mathsf{c}'\to\mathsf{f}}^{\text{SP},(\ell)}(a), \tag{3}$$

respectively. More precisely, the message $\psi_{\mathsf{c}\to\mathsf{f}}^{\text{SP},(\ell)}(a)$ represents the probability that $\phi(\mathsf{f}) = a$ given the incoming messages $\mu_{\mathsf{f}'\to\mathsf{c}}^{\text{SP},(\ell-1)}(a_{\mathsf{f}'})$ from all adjacent edges to counter node c (except the edge from f) and the constraint of the counter node itself.

Replacing the outer summation in (2) with a maximum gives the *max-product* algorithm (or, equivalently, the *min-sum* algorithm in the logarithmic domain),

$$\begin{aligned}&\psi_{\mathsf{c}\to\mathsf{f}}^{\text{MaxP},(\ell)}(a)\\ &= \max_{\substack{(a_{\mathsf{f}'}\,:\,\mathsf{f}'\in\Gamma(\mathsf{c})\backslash\mathsf{f})\,:\,a_{\mathsf{f}'}\geq f_{\min}\\ \text{and}\sum_{\mathsf{f}'\in\Gamma(\mathsf{c})\backslash\mathsf{f}} a_{\mathsf{f}'}=\phi(\mathsf{c})-a}} \prod_{\mathsf{f}'\in\Gamma(\mathsf{c})\backslash\mathsf{f}} \mu_{\mathsf{f}'\to\mathsf{c}}^{\text{MaxP},(\ell-1)}(a_{\mathsf{f}'})\end{aligned} \tag{4}$$

and

$$\mu_{\mathsf{f}\to\mathsf{c}}^{\text{MaxP},(\ell)}(a) = \prod_{\mathsf{c}'\in\Gamma(\mathsf{f})\backslash\mathsf{c}} \psi_{\mathsf{c}'\to\mathsf{f}}^{\text{MaxP},(\ell)}(a). \tag{5}$$

Both the SP algorithm and the max-product algorithm are *soft-decision* decoding algorithms which entail a high complexity.[1] To reduce decoding complexity, one may consider HDD. Assuming a uniform flow size distribution (i.e., not exploiting any information about the flow size distribution) the max-product algorithm above boils down to the low-complexity MP algorithm proposed in [1], which indeed can be seen as an HDD algorithm. In the following, we only give the update rules of the MP algorithm in [1]. Its derivation from (4) and (5) is given in Appendix A.

[1] The update rules in (2) and (4) can be implemented in each counter node using the BCJR algorithm working on a trellis where each trellis section would be a complete bipartite graph with at most $\phi(\mathsf{c}) - |\Gamma(\mathsf{c})|f_{\min} + 1$ states.

We define the messages exchanged in the MP decoding algorithm as follows. Let $\mu^{(\ell)}_{\mathsf{f}\to\mathsf{c}} \in \mathbb{N}$, where $\mathbb{N}$ is the set of natural numbers and $(\mathsf{f},\mathsf{c}) \in \mathcal{E}$, denote the message sent from flow node f to counter node c during the ℓ-th iteration of the MP decoding algorithm. Likewise, let $\psi^{(\ell)}_{\mathsf{c}\to\mathsf{f}} \in \mathbb{N}$, $(\mathsf{f},\mathsf{c}) \in \mathcal{E}$, denote the message sent from counter node c to flow node f during the ℓ-th iteration of the algorithm. The counter node and flow node update rules (for $\ell = 1,\dots,\ell_{\max}$) are given as follows [1],

$$\psi^{(\ell)}_{\mathsf{c}\to\mathsf{f}} = \max\left\{ \phi(\mathsf{c}) - \sum_{\mathsf{f}'\in\Gamma(\mathsf{c})\setminus\mathsf{f}} \mu^{(\ell-1)}_{\mathsf{f}'\to\mathsf{c}},\ f_{\min} \right\}, \tag{6}$$

$$\mu^{(\ell)}_{\mathsf{f}\to\mathsf{c}} = \begin{cases} \min_{\mathsf{c}'\in\Gamma(\mathsf{f})\setminus\mathsf{c}} \psi^{(\ell)}_{\mathsf{c}'\to\mathsf{f}}, & \text{if } \ell \text{ is odd} \\ \max_{\mathsf{c}'\in\Gamma(\mathsf{f})\setminus\mathsf{c}} \psi^{(\ell)}_{\mathsf{c}'\to\mathsf{f}}, & \text{if } \ell \text{ is even} \end{cases}, \tag{7}$$

where $\ell_{\max}$ is the maximum number of iterations and $\mu^{(0)}_{\mathsf{f}\to\mathsf{c}} = f_{\min}$ for all $(\mathsf{f},\mathsf{c}) \in \mathcal{E}$. The final estimate of the flow sizes of the flow nodes is according to

$$\hat{\phi}(\mathsf{f}) = \begin{cases} \min_{\mathsf{c}\in\Gamma(\mathsf{f})} \psi^{(\ell_{\max})}_{\mathsf{c}\to\mathsf{f}}, & \text{if } \ell_{\max} \text{ is odd} \\ \max_{\mathsf{c}\in\Gamma(\mathsf{f})} \psi^{(\ell_{\max})}_{\mathsf{c}\to\mathsf{f}}, & \text{if } \ell_{\max} \text{ is even} \end{cases}.$$

The MP algorithm in (6)–(7) can be seen as an HDD algorithm where only integer values are exchanged between flow and counter nodes. Despite this, as shown in [1, Th. 1], the output of the MP decoder converges to the exact flow size vector when the underlying CB graph is a tree.

For a multi-layer CB, decoding proceeds starting with the right-most layer. After decoding the l-th layer, represented by the bipartite graph $\mathcal{G}_l$, the counter nodes of the $(l-1)$-th layer are updated (for all $\mathsf{f} \in \mathcal{F}_l$) based on the mapping $\Xi_l(\cdot)$ as

$$\phi(\Xi_l(\mathsf{f})) \leftarrow \hat{\phi}(\mathsf{f}) \cdot 2^{d_{l-1}} + \phi(\Xi_l(\mathsf{f})).$$

Decoding proceeds layer-by-layer until the first layer is decoded.

B. Peeling Decoder

In this subsection, we introduce a peeling decoder version of the MP decoder mentioned above. The reason for doing this is that we need the concept of a residual graph and residual degree distributions in Section V.

Note that the MP decoder with the update rules in (6)–(7) will always *stop*, i.e., at some point the flow size estimates of the flow nodes will be the same at iterations ℓ and $\ell-2$ for some ℓ. This is due to the monotonicity property of the messages, i.e., for each edge $(\mathsf{f},\mathsf{c}) \in \mathcal{E}$, the messages $\mu^{(2\ell)}_{\mathsf{f}\to\mathsf{c}}$, $\ell \geq 0$, are monotonically nondecreasing lower bounds on $\phi(\mathsf{f})$, and the messages $\mu^{(2\ell+1)}_{\mathsf{f}\to\mathsf{c}}$, $\ell \geq 0$, are monotonically nonincreasing upper bounds on $\phi(\mathsf{f})$. This can be verified by induction [21]. When the decoder stops, either the flow size estimates for a flow node at iterations ℓ and $\ell-1$ are the same (i.e., we have convergence (upper and lower bounds are the same and thus we have the correct value for the flow size)), or we have oscillation (i.e., the estimates for iterations ℓ and $\ell-2$ are the same, but the estimates for iterations ℓ and $\ell-1$ are not the same).

The MP decoder can be turned into a peeling decoder in the following way. Run the MP decoder and in addition apply the following two peeling rules:[2]

1) In case $|\Gamma(\mathsf{c})| = 1$ for a counter node c, then remove c and the connected flow node $\mathsf{f} \in \Gamma(\mathsf{c})$ and all its adjacent edges from the graph. Decrease the values of the counter nodes of $\Gamma(\mathsf{f}) \setminus \mathsf{c}$ by the value of c.
2) For odd iterations, if a message from a counter node c to a flow node f is equal to $f_{\min}$, then remove the flow node f and all its adjacent edges from the graph. Decrease the values of the counter nodes of $\Gamma(\mathsf{f})$ by $f_{\min}$.

The MP and the peeling decoders are *equivalent*, in the sense that the set of *converged* flow nodes for the MP decoder, i.e., the set of flow nodes for which the flow size estimates for even and odd iterations are the same, is equal to the set of *peeled* flow nodes for the peeling decoder, i.e., the set of flow nodes that have been removed from the graph when running the peeling decoder.

C. A Note on Notation

In the rest of the paper, except for Section VI-C, we assume a single-layer system and frequently omit, for notational convenience, the subscript/superscript l. Also, we assume that the depth of the counters of the single-layer CB, d_1, is large enough so that they do not overflow. Note that if the counters overflow, one should consider multilayer CBs. This is discussed in Section VI-C.

III. Equivalent Bipartite Graph Representation of Counter Braids

In this section, we construct an *equivalent* bipartite graph representation of a single-layer CB, i.e., of the graph $\mathcal{G}$, and a corresponding MP decoding algorithm, which gives identical iteration-by-iteration finite-length and asymptotic performance to that of the original graph. The new graph has the same flow nodes as the original graph and the same number of counter nodes, but each counter node represents now a tree of depth three grown from the corresponding counter node in the original graph (a so-called computation tree of depth three, see [23, p. 27]), and the update rules are different. This graph transformation enables us to bypass the fact that to apply the potential function framework of [19], a bipartite graph with left and right update rules that do not change over the iterations is required. Note that for the original bipartite graph, this property does not hold.

The computation tree of depth three comes from the fact that, to obtain a bipartite graph with update rules that do not change with iterations, we combine two successive iterations (on the original graph) into a single one. Thus, the message from counter nodes to flow nodes of the transformed graph corresponds to the sequence (for the original graph) counter nodes – flow nodes – counter nodes – flow nodes, which leads to a computation tree of depth three.

[2] Note that we can peel all neigboring flow nodes of a counter node c if its value $\phi(\mathsf{c})$ is equal to $f_{\min}$ times its degree $\Gamma(\mathsf{c})$, i.e., if $\phi(\mathsf{c}) = f_{\min} \cdot \Gamma(\mathsf{c})$. This follows from the second peeling rule.

Algorithm 1: Construction of the equivalent graph

```
Input  : Original CB graph $\mathcal{G} = (\mathcal{F} \cup \mathcal{C}, \mathcal{E})$
Output: Equivalent CB graph $\tilde{\mathcal{G}} = \left(\tilde{\mathcal{F}} \cup \tilde{\mathcal{C}}, \tilde{\mathcal{E}}\right)$
$\tilde{\mathcal{F}} \leftarrow \emptyset$                        ▷ $\tilde{\mathcal{F}}$ is a set
$\tilde{\mathcal{C}} \leftarrow \emptyset$                        ▷ $\tilde{\mathcal{C}}$ is a set
$\tilde{\mathcal{E}} \leftarrow \emptyset$                        ▷ $\tilde{\mathcal{E}}$ is a multiset
for $\mathsf{f} \in \mathcal{F}$ do
    $\tilde{\mathcal{F}} \leftarrow \mathsf{f}$
end
for $\mathsf{c} \in \mathcal{C}$ do
    $\tilde{\mathcal{C}} \leftarrow \mathsf{c}$
    for $\mathsf{f} \in \mathcal{F}_1(\mathsf{c}) \cup \mathcal{F}_3(\mathsf{c})$ do
        $\tilde{\mathcal{E}} \leftarrow (\mathsf{c}, \mathsf{f})$ with label equal to the length of
        $\mathcal{P}(\mathsf{c}, \mathsf{f})$
    end
end
return $\left(\tilde{\mathcal{F}} \cup \tilde{\mathcal{C}}, \tilde{\mathcal{E}}\right)$
```

For simplicity, we will first consider the example graph $\mathcal{G}$ shown in Fig. 2(a) for the construction of the equivalent graph in Section III-A, before presenting the general construction algorithm (Algorithm 1). The MP decoder for the equivalent graph is described in Section III-B for the example graph $\tilde{\mathcal{G}}$ shown in Fig. 2(c), from which the general case follows in a straightforward manner.

A. Construction

The equivalent graph, denoted by $\tilde{\mathcal{G}}$, is constructed as follows for the example graph $\mathcal{G}$ shown in Fig. 2(a). First, for each counter node $\mathsf{c} \in \mathcal{C}$ in the graph in Fig. 2(a), we build a (computation) tree $\mathcal{T}(\mathsf{c})$ of depth three [23, p. 27], as illustrated in Fig. 2(b) for the counter node c_1 (for notational convenience this counter node is labeled with the integer 1 instead of c_1 in the figure). Then, for each counter node $\mathsf{c} \in \mathcal{C}$ we make a new counter node $\tilde{\mathsf{c}}$ in the new bipartite graph $\tilde{\mathcal{G}}$ (see Fig. 2(c)). Furthermore, in the new graph, we make a copy $\tilde{\mathsf{f}}$ of each flow node $\mathsf{f} \in \mathcal{F}$ in the original graph. Counter nodes and flow nodes in the new graph are connected using two types of edges, type-1 and type-2, as explained in the following. Consider the counter node c_1 in Fig. 2(a). For this counter node, we make a new counter node $\tilde{\mathsf{c}}_1$ in the equivalent graph in Fig. 2(c). Then, we connect $\tilde{\mathsf{c}}_1$ to a flow node $\tilde{\mathsf{f}}_i$ with a type-2 edge if f_i is a leaf in the tree of c_1. Furthermore, we connect $\tilde{\mathsf{c}}_1$ to a flow node $\tilde{\mathsf{f}}_i$ with a type-1 edge if f_i is a flow node at depth one in the tree of c_1. In this way there is an edge for each path in $\mathcal{T}(\mathsf{c}_1)$ from c_1 to a flow node. When the path has length three, the edge is a type-2 edge, and when the path has length one, the edge is a type-1 edge.

In Fig. 2(c), type-1 edges are represented by nonbold edges and correspond each to a distinct path of length one from c_1 or c_2 to a flow node at depth one in $\mathcal{T}(\mathsf{c}_1)$ or $\mathcal{T}(\mathsf{c}_2)$, respectively. On the other hand, bold edges represent edge bundles of type-2 edges. For the example in Fig. 2(c), an edge bundle between counter node $\tilde{\mathsf{c}}_1$ and flow node $\tilde{\mathsf{f}}_i$, $i = 1, 2, 3, 4$, consists of three type-2 edges corresponding to the three distinct paths of length three in the tree $\mathcal{T}(\mathsf{c}_1)$ between c_1 and flow node f_i, at depth three. For notational convenience, these flow nodes are labeled with the integers i instead of $\tilde{\mathsf{f}}_i$ in Fig. 2(c) and instead of f_i in Figs. 2(a) and 2(b). Likewise, $\tilde{\mathsf{c}}_1$ is labeled by the integer 1 in Fig. 2(c). Note that for a given bipartite graph $\mathcal{G}$, the number of type-2 edges in each edge bundle of the equivalent bipartite graph $\tilde{\mathcal{G}}$ depends on the connectivity of $\mathcal{G}$. Also, in the asymptotic limit there will be no duplicate flow nodes in the trees $\mathcal{T}(\mathsf{c})$ for any counter node $\mathsf{c} \in \mathcal{C}$ and the equivalent graph will be locally tree-like.

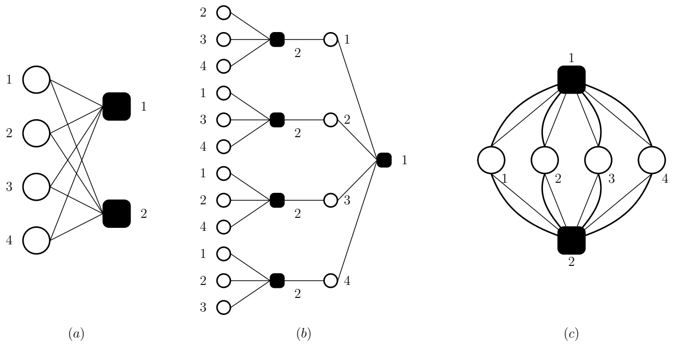

Fig. 2. (a) The graph $\mathcal{G}$ of an example CB. (b) The tree $\mathcal{T}(\mathsf{c}_1)$ of depth three grown from the counter node c_1 (for notational convenience the counter node is labeled with the integer 1 instead of c_1 in the figure) from the original graph $\mathcal{G}$ in (a). (c) Equivalent bipartite graph representation $\tilde{\mathcal{G}}$ of the example CB from (a).

For the general case, let $\mathcal{T}(\mathsf{c}) = (\mathcal{V}(\mathsf{c}), \mathcal{E}(\mathsf{c}))$ denote the (computation) tree for node c, where $\mathcal{V}(\mathsf{c}) = \mathsf{c} \cup \mathcal{F}_1(\mathsf{c}) \cup \mathcal{C}_2(\mathsf{c}) \cup \mathcal{F}_3(\mathsf{c})$, $\mathcal{F}_1(\mathsf{c})$ is the set of neighboring flow nodes of c (distance-1 nodes from c), $\mathcal{C}_2(\mathsf{c})$ is the set of distance-2 nodes from c (all counter nodes), and $\mathcal{F}_3(\mathsf{c})$ is the set of distance-3 nodes from c (all flow nodes). The set of edges of the computation tree is denoted by $\mathcal{E}(\mathsf{c})$. For notational convenience, the (unique) path between two nodes $\mathsf{v}_1 \in \mathcal{V}(\mathsf{c})$ and $\mathsf{v}_2 \in \mathcal{V}(\mathsf{c})$ is denoted by $\mathcal{P}(\mathsf{v}_1, \mathsf{v}_2)$. The construction of the equivalent graph $\tilde{\mathcal{G}} = (\tilde{\mathcal{F}} \cup \tilde{\mathcal{C}}, \tilde{\mathcal{E}})$, which is a graph with parallel edges (or a multigraph), can be described by Algorithm 1. For a multigraph, the edge set $\tilde{\mathcal{E}}$ is a multiset, i.e., unlike a set, it allows multiple instances of the elements of the multiset. If the label of an edge in $\tilde{\mathcal{G}}$ is 1, then the edge is a type-1 edge. Otherwise, it is a type-2 edge. Since the original CB graph $\mathcal{G}$ does not contain parallel edges, there are no parallel type-1 edges in $\tilde{\mathcal{G}}$. However, there might be parallel type-2 edges, and these can be grouped together in edge bundles, where each edge bundle contains all parallel edges between a given pair of endpoints.

B. Message Passing Decoding

We have the following update rules for the new graph. For the flow nodes $\tilde{\mathsf{f}} \in \tilde{\mathcal{F}}$ we use the same update rule as in (7) for type-1 edges, using the maximum only, over all incoming messages on all connected edges (except the one on which the outgoing message is transmitted). For the type-2 edges in an edge bundle, the update rule is a bit different. Note that each edge bundle is in one-to-one correspondence with paths of length three from the root node in the tree to the leaf nodes. Thus, we can label each type-2 edge in the bundle from flow node $\tilde{\mathsf{f}}$ to counter node $\tilde{\mathsf{c}}$ with the counter node in the new graph that corresponds to the parent counter node of the corresponding leaf node f in the tree. Then, the output message on a type-2 edge in the bundle will be the same as the output message on the type-1 edge of $\tilde{\mathcal{G}}$ that connects flow node $\tilde{\mathsf{f}}$ to the counter node that labels the type-2 edge in the bundle which is being considered. Equivalently, we can use the update rule in (7), using the maximum only, over all incoming messages on all connected edges except the particular type-1 edge connected to the counter node that labels the type-2 edge in the bundle under consideration.

For the counter nodes $\tilde{\mathsf{c}} \in \tilde{\mathcal{C}}$ in the new graph, the update rule is a bit more complicated and the corresponding tree $\mathcal{T}(\mathsf{c})$ must be used. The procedure consists of three steps as follows:

1) First, the outgoing message on an edge from a flow node $\tilde{\mathsf{f}}$ to a counter node $\tilde{\mathsf{c}}$ in the new graph is assigned as an outgoing message to the corresponding edge emanating from the corresponding node f in the tree $\mathcal{T}(\mathsf{c})$. Then, run the update rule in (6) for all counter nodes in the tree $\mathcal{T}(\mathsf{c})$ using all incoming messages (except the one on which the outgoing message is transmitted). Outgoing messages are sent to all connected flow nodes at depth one in the tree $\mathcal{T}(\mathsf{c})$ (i.e., upward towards the root node).
2) Secondly, run the update rule in (7) for all flow nodes at depth one in the tree $\mathcal{T}(\mathsf{c})$ (using the minimum only) using all incoming messages (except the one on which the outgoing message is transmitted). Outgoing messages are sent only to the root node c in the tree $\mathcal{T}(\mathsf{c})$ (i.e., upward to the root node).
3) Finally, apply the update rule in (6) on the root node using all incoming messages (except the one on which the outgoing message is transmitted). Outgoing messages to all connected flow nodes at depth one are computed and assigned to the corresponding type-1 edges in the new graph, i.e., a message from the root node c to a flow node f at depth one is assigned as an outgoing message on a type-1 edge from counter node $\tilde{\mathsf{c}}$ to flow node $\tilde{\mathsf{f}}$ in the new graph $\tilde{\mathcal{G}}$.

Note that the three-step procedure above only assigns outgoing messages to type-1 edges connected to $\tilde{\mathsf{c}}$ in the new graph. The outgoing messages on the type-2 edges in the edge bundles are set to $-\infty$.

Decoding on the equivalent graph $\tilde{\mathcal{G}}$ is performed as in the standard flooding schedule, i.e., all counter nodes in $\tilde{\mathcal{G}}$ should be processed as outlined above before processing all flow nodes. On the other hand, encoding can be done using the equivalent graph as well by ignoring the type-2 edges and using the encoding method of the original graph, i.e., the value of a counter node is the sum modulo 2^{d_1} of the values of the connected flow nodes (considering only type-1 edges). Notice that for any CB graph $\mathcal{G}$ we can construct an equivalent graph $\tilde{\mathcal{G}}$ using the construction above, and viceversa (removing the edge bundles in $\tilde{\mathcal{G}}$).

Let $\hat{\phi}_{\mathcal{G}}^{(\ell)}(\mathsf{f})$ denote the estimated flow size at flow node $\mathsf{f} \in \mathcal{G}$ at iteration ℓ under MP decoding on the original CB graph $\mathcal{G}$ (see Section II-A). The corresponding estimated flow size at iteration ℓ and flow node $\tilde{\mathsf{f}} \in \tilde{\mathcal{G}}$ (corresponding to flow node f in the original graph $\mathcal{G}$) under MP decoding on the equivalent graph $\tilde{\mathcal{G}}$ (as described above) is denoted by $\hat{\phi}_{\tilde{\mathcal{G}}}^{(\ell)}(\tilde{\mathsf{f}})$. Based on the above discussion, the proposition below follows by inspection.

Proposition 1. *For any given CB graph $\mathcal{G}$ and iteration number $\ell \geq 1$ (with respect to the original graph $\mathcal{G}$),*

$$\hat{\phi}_{\mathcal{G}}^{(2\ell)}(\mathsf{f}) = \hat{\phi}_{\tilde{\mathcal{G}}}^{(\ell)}(\tilde{\mathsf{f}}),\ \forall \mathsf{f} \in \mathcal{G},$$

i.e., the two graphs are iteration-by-iteration *equivalent.*

IV. Asymptotic Analysis of Single-Layer Counter Braids

Consider a single-layer CB with m_0 flow nodes and m_1 counter nodes of depth d_1. When $m_0 \to \infty$, the asymptotic performance of CBs can be analyzed by means of DE. Denote by $x^{(\ell)}$ and $y^{(\ell)}$ the error probability of an outgoing message from a flow node and a counter node, respectively, at the ℓ-th iteration. The flow node and counter node DE updates at iteration ℓ are described by the equations [1], [2]

$$x^{(\ell)} = \tilde{f}\left(y^{(\ell)}; \epsilon\right), \quad y^{(\ell)} = \tilde{g}\left(x^{(\ell-1)}\right), \tag{8}$$

where

$$\tilde{f}(y;\epsilon) = \begin{cases} y^{k-1}, & \text{if } \ell \text{ is odd} \\ \epsilon \cdot y^{k-1}, & \text{if } \ell \text{ is even} \end{cases}, \tag{9}$$

$$\tilde{g}(x) = 1 - \rho\left(1 - x\right), \tag{10}$$

and ϵ is the probability of observing a flow of size strictly larger than $f_{\min}$. Note that ϵ depends on the flow size distribution, denoted by p_Φ.

In the following, we denote by $\mathcal{X} = [0,1]$ and $\mathcal{Y} = [0,1]$ the set of possible values for x and y, respectively, and by $\mathcal{E}$ the set of possible values for ϵ.

For a given flow size distribution, or more precisely, for a given ϵ, a relevant parameter for the performance of single-layer CBs is the number of counters per flow, $\beta = \frac{m_1}{m_0} = \frac{k}{\gamma}$. The average number of bits needed to represent a flow is therefore βd_1, hence β is directly related to the compression rate. In particular, we are interested in the minimum value of β so that decoding is successful,

$$\beta_{\rm MP} = \beta_{\rm MP}(\epsilon) \triangleq \inf\left\{\beta > 0 \mid x^{(\infty)}(\beta,\epsilon) = 0\right\}.$$

Alternatively, we can analyze the asymptotic behavior of CBs by fixing β and finding the maximum value of ϵ such that decoding is successful,

$$\epsilon_{\rm MP} = \epsilon_{\rm MP}(\beta) \triangleq \sup\left\{\epsilon \in \mathcal{E} \mid x^{(\infty)}(\beta,\epsilon) = 0\right\},$$

since for a fixed k and any ϵ it follows that $\epsilon_{\rm MP}(\beta_{\rm MP}(\epsilon)) = \beta_{\rm MP}^{-1}(\beta_{\rm MP}(\epsilon)) = \epsilon$. In this case, $\epsilon_{\rm MP}$ has a similar meaning as the MP decoding threshold for LDPC codes over the binary erasure channel (BEC), where ϵ can now be interpreted as the channel parameter.

Combining two successive iterations into a single one, we can rewrite the DE in an equivalent form as

$$x^{(2\ell)} = f\left(y^{(2\ell)}; \epsilon\right), \quad y^{(2\ell)} = g\left(x^{(2\ell-2)}\right), \tag{11}$$

where

$$f(y;\epsilon) = \epsilon \cdot y^{k-1}, \tag{12}$$

$$g(x) = 1 - \rho\left(1 - (1-\rho(1-x))^{k-1}\right). \tag{13}$$

Clearly, the DE recursions in (8)–(10) and (11)–(13) give the same MP decoding threshold. Furthermore, it is easy to verify that the DE in (11)–(13) corresponds to the DE of the equivalent graph introduced in the previous section. Therefore, from this and Proposition 1, we can analyze CBs based on the equivalent graph and the DE recursion in (11)–(13).

Lemma 1. *The functions $f(y;\epsilon)$ (for a fixed value of $\epsilon > 0$) and $g(x)$, $k \geq 2$, are strictly increasing in y and x, respectively.*

Proof: From the definition above it follows directly that $f(y;\epsilon)$ is strictly increasing in y for a fixed value of $\epsilon > 0$. Also, since $\rho(x)$ is strictly increasing in x and both $1-\rho(x)$ and $\rho(1-x)$ are strictly decreasing in x, it follows directly that $g(x)$ is strictly increasing in x. □

As a result of Lemma 1, it follows that the DE recursion in (11)–(13) (and thus the DE recursion in (8)–(10)) for uncoupled CBs converges to a fixed-point. The fixed-point DE equation for $x = x^{(\infty)}(\beta,\epsilon)$ is

$$x = f(g(x);\epsilon). \tag{14}$$

Decoding is successful if the fixed-point is $x^{(\infty)}(\beta,\epsilon) = 0$.

We will need the definition of an *admissible system* [19].

Definition 1 (Admissible System)**.** *An admissible system is a system where the functions $f(y;\epsilon)$ and $g(x)$ satisfy the following properties.*

1) *The first derivative of $f(y;\epsilon)$ exists and is continuous on $\mathcal{Y}\times\mathcal{E}$, the first derivative of $g(x)$ exists and is continuous on $\mathcal{X}$,*
2) *$f(y;\epsilon)$ is nondecreasing in both y and ϵ,*
3) *$g(x)$ is strictly increasing in x, and*
4) *the second derivative of $g(x)$ exists and is continuous on $\mathcal{X}$.*

In addition, an admissible system is said to be *proper* if the derivative of $h(x;\epsilon) \triangleq f(g(x);\epsilon)$ with respect to ϵ is strictly positive for all $(x,\epsilon) \in (0,1] \times \mathcal{E}$ [19].

Lemma 2. *The system defined in (12)–(13) is a proper admissible system.*

Proof: It is easy to show that the DE updates in (12)–(13) satisfy Properties 1 to 4 in Definition 1. Properties 2 and 3 have already been proven in Lemma 1. Now, the derivative of $h(x;\epsilon)$ with respect to ϵ is

$$\left(1-\rho\left(1-(1-\rho(1-x))^{k-1}\right)\right)^{k-1},$$

and the result follows from the fact that $\rho(x)$ for $x \in (0,1]$ and $k \geq 2$ is strictly positive. □

A. Extended Message Passing Extrinsic Information Transfer Curve

The EMP EXIT curve of a single-layer CB is given in parametric form by

$$(\epsilon, h^{\mathrm{EMP}}) = (\epsilon(x), (1-\rho(1-(1-\rho(1-x))^{k-1}))^{k}), \tag{15}$$

where

$$\epsilon(x) = \frac{x}{(1-\rho(1-(1-\rho(1-x))^{k-1}))^{k-1}}$$

is the solution of (14) for ϵ, and $x \in \mathcal{X}$. The curve is a trace of all fixed-points of the DE recursion in (14) (both stable and unstable).

We have the following theorem related to the EMP EXIT curve, adapting [20, Th. 8] to the recursion in (12)–(13).

Theorem 1. *The EMP EXIT curve satisfies*

$$\int_0^1 h^{\mathrm{EMP}}(x)\ \mathrm{d}\epsilon(x) =$$
$$1-\rho(1-(1-\mathrm{e}^{-\gamma})^{k-1}) + k\left(\rho(1-(1-\mathrm{e}^{-\gamma})^{k-1}) - \int_0^1 \rho(1-(1-\rho(1-z))^{k-1})\ \mathrm{d}z\right).$$

Proof: The result follows by applying integration by parts twice as in the proof of [20, Th. 8]. Details are omitted for brevity. □

Note that, contrary to the case of standard (both regular and irregular) LDPC code ensembles on the BEC, the area is *not* equal to the rate (in number of counters per flow) and the curve does *not* start at the $(1,1)$ point for $x = 1$. In fact, the EMP EXIT curve starts (for $x = 1$) at the point $(1/(1-\rho(1-(1-\mathrm{e}^{-\gamma})^{k-1}))^{k-1}, (1-\rho(1-(1-\mathrm{e}^{-\gamma})^{k-1}))^{k})$, where the first coordinate is larger than 1 and the second coordinate is smaller than 1. Also, as x approaches 0, when $k \geq 3$, the curves approach the point $(\infty, 0)$ (as could also be the case for LDPC code ensembles).

Definition 2 (Area Threshold)**.** *Let $(\epsilon(x^*), h^{\mathrm{EMP}}(x^*))$ be a point on the EMP EXIT curve h^{EMP} of a single-layer CB such that*

$$\int_{x^*}^1 h^{\mathrm{EMP}}(x)\ \mathrm{d}\epsilon(x) = \int_0^1 h^{\mathrm{EMP}}(x)\ \mathrm{d}\epsilon(x)$$

and assume that there exist no $x' \in (x^*, 1]$ such that $\epsilon(x') = \epsilon(x^*)$. Then, the area threshold, denoted by $\bar{\epsilon}$, is defined as $\bar{\epsilon} = \epsilon(x^*)$.

Lemma 3. *For $k = 2$ the area threshold is equal to $\bar{\epsilon} = 1/\gamma^2$.*

Proof: For $k = 2$, $\epsilon(x) = \frac{x}{1-\rho(\rho(1-x))}$ is an increasing function of x. Thus, $x^* = 0$, and the area threshold becomes

$$\bar{\epsilon} = \lim_{x\to 0} \frac{x}{1-\rho(\rho(1-x))} = \frac{1}{\rho'(\rho(1-0))\rho'(1-0)}$$
$$= \frac{1}{\gamma\rho(\rho(1))\gamma\rho(1)} = \frac{1}{\gamma^2},$$

where $\rho'(x)$ denotes the derivative of $\rho(x)$ with respect to x, and the result follows. □

Note that this threshold corresponds to the *stability* threshold from the theory of LDPC codes on the BEC.

In Section V, we will prove that the area threshold $\bar{\epsilon}$ is an upper bound on the Maxwell decoding threshold.

B. Potential Function, Potential Threshold, and Area Threshold

Since the DE recursion in (11)–(13) describes an admissible system (see Definition 1), we can define a corresponding potential function as in [19].

Definition 3. *The potential function $U(x;\epsilon)$ of the system defined by the functions f and g from (12) and (13), respectively, is given by*

$$\begin{aligned} U(x;\epsilon) &\triangleq xg(x) - \int_0^x g(z)\ \mathrm{d}z - \int_0^{g(x)} f(z;\epsilon)\ \mathrm{d}z \\ &= xg(x) - \int_0^x g(z)\ \mathrm{d}z - \frac{\epsilon}{k} g(x)^k. \end{aligned}$$

Following [19, Def. 32], we make the following definitions,

$$\begin{aligned} \Psi(\epsilon) &\triangleq \min_{x\in\mathcal{X}} U(x;\epsilon), \\ X^*(\epsilon) &\triangleq \{x \in \mathcal{X} \mid U(x;\epsilon) = \Psi(\epsilon)\}, \text{ and} \\ \bar{x}^*(\epsilon) &\triangleq \max X^*(\epsilon). \end{aligned}$$

Now, we can define the *potential threshold* ϵ_{p}^* as [19, Def. 35]

$$\epsilon_{\mathrm{p}}^* \triangleq \sup\left\{\epsilon \in \mathcal{E} \mid \bar{x}^*(\epsilon) = 0\right\}. \tag{16}$$

Definition 4. *The* fixed-point potential $Q(x)$ *is given by [19, Def. 41]*

$$\begin{aligned} Q(x) &\triangleq U(x;\epsilon(x)) \\ &= xg(x) - \int_0^x g(z)\ \mathrm{d}z - \frac{1}{k}\cdot\frac{x}{g(x)^{k-1}}\cdot g(x)^k \\ &= \left(1-\frac{1}{k}\right)xg(x) - \int_0^x g(z)\ \mathrm{d}z. \end{aligned}$$

Definition 5. *The* trial entropy *is given by*

$$P(x) \triangleq \int_0^x h^{\mathrm{EMP}}(z)\epsilon'(z)\ \mathrm{d}z = \int_0^x g(z)^k\epsilon'(z)\ \mathrm{d}z.$$

The following theorem shows that the area threshold and the potential threshold are equal, $\bar{\epsilon} = \epsilon_{\mathrm{p}}^*$.

Theorem 2. *The area threshold from Definition 2 is equal to the potential threshold from (16).*

Proof: It follows that

$$\begin{aligned} g(z)^k\epsilon'(z) &= g(z)^k\left(\frac{g(z)^{k-1} - (k-1)g(z)^{k-2}g'(z)z}{g(z)^{2(k-1)}}\right) \\ &= g(z) - (k-1)zg'(z). \end{aligned}$$

Thus,

$$\begin{aligned} P(x) &= \int_0^x g(z)\ \mathrm{d}z - (k-1)\int_0^x zg'(z)\ \mathrm{d}z \\ &= \int_0^x g(z)\ \mathrm{d}z - (k-1)\left(zg(z)\Big|_0^x - \int_0^x g(z)\ \mathrm{d}z\right) \\ &= k\int_0^x g(z)\ \mathrm{d}z + (1-k)xg(x) \\ &= -kQ(x). \end{aligned} \tag{17}$$

We have

$$\begin{aligned} \int_x^1 h^{\mathrm{EMP}}(z)\ \mathrm{d}\epsilon(z) &= \int_0^1 h^{\mathrm{EMP}}(z)\ \mathrm{d}\epsilon(z) \\ &\quad - \int_0^x h^{\mathrm{EMP}}(z)\ \mathrm{d}\epsilon(z) \\ &\overset{(a)}{=} \int_0^1 h^{\mathrm{EMP}}(z)\ \mathrm{d}\epsilon(z) - P(x), \end{aligned} \tag{18}$$

where (a) follows from the definition of the trial entropy in Definition 5. Now, using Definition 2 it follows that $P(x^*) = 0$. Using (17) this implies that $Q(x^*) = 0$. From [19, Lem. 46], for a proper admissible system, which is the case here as shown in Lemma 2, the corresponding value $\bar{\epsilon} = \epsilon(x^*)$ is indeed equal to the potential threshold, and the result follows. □

In the next section, we derive the Maxwell decoder for CBs and prove that the area threshold $\bar{\epsilon}$ is an upper bound on the Maxwell decoding threshold (formally defined below), which, in turn, is a lower bound on the MAP decoding threshold (which is also formally defined below). We then give a conjecture, supported by simulation results, that the area threshold is in fact equal to the Maxwell decoding threshold, and thus a lower bound on the MAP decoding threshold.

V. Maxwell Decoder

Similar to LDPC codes, a Maxwell decoder [20] can be constructed for CBs, and it can be analyzed asymptotically using DE on the equivalent graph representation introduced in Section III. For LDPC codes, the Maxwell decoder is a MAP decoder, thus the MAP decoding threshold is equal to the Maxwell decoding threshold. However, it is important to notice that, in general, for CBs the Maxwell decoder is *not* a MAP decoder, since the flow size distribution is typically nonuniform. Thus, the Maxwell decoding threshold (on ϵ) is in general a lower bound on the MAP decoding threshold.

The MAP EXIT curve of a single-layer CB is defined as

$$h^{\mathrm{MAP}}(\epsilon) = \limsup_{m_0\to\infty} \mathbb{E}\left[\frac{1}{m_0}\sum_{\mathsf{f}\in\mathcal{F}} \mathbb{H}(\phi(\mathsf{f})|\{\phi(\mathsf{c}) : \mathsf{c}\in\mathcal{C}\})\right],$$

where $\mathbb{H}(\cdot|\cdot)$ denotes conditional entropy and the expectation $\mathbb{E}[\cdot]$ is taken with respect to the ensemble of bipartite graphs $\mathcal{G}$ of the underlying CBs. The MAP decoding threshold, denoted by ϵ_{MAP}, is defined as

$$\epsilon_{\mathrm{MAP}} \triangleq \sup\left\{\epsilon \in \mathcal{E} \mid h^{\mathrm{MAP}}(\epsilon) = 0\right\}.$$

The Maxwell decoding threshold (and EXIT chart) can be defined in an analogue manner under the constraint of a uniform flow size distribution.

A. Maxwell Decoder on the Original Graph

The Maxwell decoder is an MP decoder with guesses, inspired from statistical mechanics and the theory of phase transitions, which was outlined in [20] in the context of LDPC codes. The concept of an MP decoder with guesses was considered in various works prior to [20], with the more practical motivation of improving the performance of the standard MP decoder for finite-length codes [24], [25]. The motivation in [20], on the contrary, was an asymptotic analysis.

The Maxwell decoder works in the following way. First, standard MP decoding as described in Section II-A is run until the algorithm stops, i.e., there is no further progress from one iteration to the next one. Then, the value of an *unknown* (to be defined below) flow node is *guessed* and all its outgoing messages (more precisely, the second component of the messages, see below for details) are set to 'g'. Moreover, the label of the flow node is changed from *unknown* to *guessed*. Decoding is performed again until there is no further progress. If there are still *unknown* flow nodes when the algorithm stops, a second *unknown* flow node is *guessed*, its label is changed to *guessed* and, finally, all its outgoing messages (the second component of the messages) are set to 'g'. This process continues until there is no more *unknown* flow nodes to guess. In the end we end up with a system of linear equations for the values of the *guessed* flow nodes that can be solved. The system of equations may or may not have a unique solution. In case there is no unique solution, ML decoding will also fail.

Let $\boldsymbol{\psi}^{(\ell),\mathrm{M}}_{\mathsf{c}\to\mathsf{f}}$ denote the message from counter node c to flow node f at iteration $\ell \geq 1$ of the Maxwell decoder. Likewise, let $\boldsymbol{\mu}^{(\ell),\mathrm{M}}_{\mathsf{f}\to\mathsf{c}}$ denote the corresponding message from flow node f to counter node c at iteration $\ell \geq 0$. The messages in the Maxwell decoder are 2-tuples defined as

$$\begin{aligned}\boldsymbol{\psi}^{(\ell),\mathrm{M}}_{\mathsf{c}\to\mathsf{f}} &= (\psi^{(\ell),\mathrm{M}}_{1,\mathsf{c}\to\mathsf{f}}, \psi^{(\ell),\mathrm{M}}_{2,\mathsf{c}\to\mathsf{f}}),\\ \boldsymbol{\mu}^{(\ell),\mathrm{M}}_{\mathsf{f}\to\mathsf{c}} &= (\mu^{(\ell),\mathrm{M}}_{1,\mathsf{f}\to\mathsf{c}}, \mu^{(\ell),\mathrm{M}}_{2,\mathsf{f}\to\mathsf{c}}).\end{aligned}$$

The first component of the messages is updated according to the update rules of the standard MP decoder described in Section II-A. The second component of the messages takes values in the set $\{0, *, \mathrm{g}\}$. The meaning of a 0-message is that the value of the flow node from which this message emanates is *known*, the meaning of the $*$-message is that the value of the flow node is *unknown* and, finally, the meaning of a g-message is that the value of the flow node has been guessed or that the value of the flow node can be expressed as a linear combination of the values of other flow nodes that all have been guessed. Operationally, we can think of a g-message emanating from a flow node f as a shorthand notation for a nonempty list $\{\mathsf{f}_{j_1}, \ldots, \mathsf{f}_{j_l}, \mathsf{b}_{j_1}, \ldots, \mathsf{b}_{j_l}, \mathsf{K}\}$, where $\{\mathsf{f}_{j_1}, \ldots, \mathsf{f}_{j_l}\}$ is a list of flow nodes whose values have been guessed, $\{\mathsf{b}_{j_1}, \ldots, \mathsf{b}_{j_l}\}$, with $\mathsf{b}_{j_i} \in \mathbb{Z}$, where $\mathbb{Z}$ is the set of integers, is a corresponding list of integer coefficients, and K is an integer that indicates that $\phi(\mathsf{f})$ can be expressed as $\phi(\mathsf{f}) = \mathsf{K} - \mathsf{b}_{j_1}\phi(\mathsf{f}_{j_1}) - \cdots - \mathsf{b}_{j_l}\phi(\mathsf{f}_{j_l})$.

For the second component of the counter-to-flow node messages we have the following update rule, for $\ell = 1, \ldots, \ell_{\max}$,

$$\psi^{(\ell),\mathrm{M}}_{2,\mathsf{c}\to\mathsf{f}} = \begin{cases} 0, & \text{if } \forall \mathsf{f}' \in \Gamma(\mathsf{c}) \setminus \mathsf{f},\ \mu^{(\ell-1),\mathrm{M}}_{2,\mathsf{f}'\to\mathsf{c}} = 0 \\ *, & \text{if } \exists \mathsf{f}' \in \Gamma(\mathsf{c}) \setminus \mathsf{f},\ \mu^{(\ell-1),\mathrm{M}}_{2,\mathsf{f}'\to\mathsf{c}} = * \\ \mathrm{g}, & \text{if } \forall \mathsf{f}' \in \Gamma(\mathsf{c}) \setminus \mathsf{f},\ \mu^{(\ell-1),\mathrm{M}}_{2,\mathsf{f}'\to\mathsf{c}} \neq * \text{ and} \\ & \quad \exists \mathsf{f}' \in \Gamma(\mathsf{c}) \setminus \mathsf{f},\ \mu^{(\ell-1),\mathrm{M}}_{2,\mathsf{f}'\to\mathsf{c}} = \mathrm{g} \end{cases}. \tag{19}$$

Initially, we set $\mu^{(0),\mathrm{M}}_{2,\mathsf{f}\to\mathsf{c}} = *$ for all $(\mathsf{f}, \mathsf{c}) \in \mathcal{E}$. The last rule in (19) states that we will get a g-message if for all connected flow nodes (except the one to which the outgoing message is transmitted) the value is either known, has been guessed, or can be expressed as a linear combination of the values of guessed flow nodes, and at least one connected flow node sends a g-message. Since the outgoing message from a counter node is the difference of the value of the counter node and the sum of the values of all its neighboring flow nodes (except the one to which the outgoing message is transmitted) (see (6)), it follows that the outgoing message can also be expressed as a linear combination of guessed flow node values. Operationally, we have several lists entering the counter node c. The outgoing list of flow nodes will be the *union* of the incoming lists, where the coefficient of flow nodes that occur a multiple number of times in the incoming lists is replaced by the corresponding sum of coefficients multiplied by -1. If the sum is equal to zero, the flow node is eliminated from the outgoing list. Also, the integer K of the outgoing list is $\phi(\mathsf{c})$ minus the sum of all K's of the incoming lists.

For the flow-to-counter node messages, the first component is again updated according to the update rule of the standard MP decoder described in Section II-A. For the second component, we have the following update rule, for $\ell = 1, \ldots, \ell_{\max}$,

$$\mu^{(\ell),\mathrm{M}}_{2,\mathsf{f}\to\mathsf{c}} = \begin{cases} 0, & \text{if } \exists \mathsf{c}' \in \Gamma(\mathsf{f}) \setminus \mathsf{c},\ \psi^{(\ell),\mathrm{M}}_{2,\mathsf{c}'\to\mathsf{f}} = 0 \text{ or} \\ & \quad \mu^{(\ell),\mathrm{M}}_{1,\mathsf{f}\to\mathsf{c}} = \mu^{(\ell-1),\mathrm{M}}_{1,\mathsf{f}\to\mathsf{c}} \\ *, & \text{if } \forall \mathsf{c}' \in \Gamma(\mathsf{f}) \setminus \mathsf{c},\ \psi^{(\ell),\mathrm{M}}_{2,\mathsf{c}'\to\mathsf{f}} = * \\ \mathrm{g}, & \text{if } \forall \mathsf{c}' \in \Gamma(\mathsf{f}) \setminus \mathsf{c},\ \psi^{(\ell),\mathrm{M}}_{2,\mathsf{c}'\to\mathsf{f}} \neq 0 \text{ and} \\ & \quad \exists \mathsf{c}' \in \Gamma(\mathsf{f}) \setminus \mathsf{c},\ \psi^{(\ell),\mathrm{M}}_{2,\mathsf{c}'\to\mathsf{f}} = \mathrm{g} \end{cases}, \tag{20}$$

where we assume that the first component of the messages, i.e., $\mu^{(\ell),\mathrm{M}}_{1,\mathsf{f}\to\mathsf{c}}$ is updated first.

B. Maxwell Decoder on the Equivalent Graph

The Maxwell decoder can also be implemented on the equivalent graph introduced in Section III. In this case, the first component of the messages is updated according to the update rules of the MP decoder on the equivalent graph, as described in Section III-B. The update rules for the second component are as follows.

For the flow nodes $\tilde{\mathsf{f}} \in \tilde{\mathcal{F}}$ in the equivalent graph, we use the same update rule as in (20) for type-1 edges over all incoming messages on all connected edges (except the one on which the outgoing message is transmitted). On the other hand, for type-2 edges in an edge bundle, the update rule is a bit different. Similar to the MP decoder (see Section III), we can use the same update rule as for type-1 edges, i.e., the update rule in (20), applied over all incoming messages on all connected edges except the particular type-1 edge connected to the counter node that labels the type-2 edge in the bundle under consideration.

For the counter nodes $\tilde{\mathsf{c}} \in \tilde{\mathcal{C}}$ in the equivalent graph, the update rule follows a three-step procedure, similar to that of the MP decoder described in Section III-B. In particular, the three-step procedure is the same as for the MP decoder, with the difference that the update rule in (19), the update rule in (20), and the update rule in (19) are used in steps 1, 2, and 3, respectively, of the three-step procedure described in

Section III-B (instead of (6), (7), and (6), respectively, for the MP decoder). Furthermore, as for the MP decoder on the equivalent graph, this three-step procedure only assigns outgoing messages on the type-1 edges connected to $\tilde{\mathsf{c}}$ in the equivalent graph. The second component of the outgoing messages on the type-2 edges in the edge bundles is set to $*$.

Now, let $\hat{\phi}_{\mathcal{G}}^{(\ell),\mathrm{M}}(\mathsf{f})$ denote the estimated flow size at flow node $\mathsf{f} \in \mathcal{G}$ at iteration ℓ under Maxwell decoding on the original CB graph $\mathcal{G}$ and $\mathsf{l}_{\mathcal{G}}^{(\ell),\mathrm{M}}(\mathsf{f})$ its label ("guessed", "unknown", or "known"). The corresponding estimated flow size at iteration ℓ and flow node $\tilde{\mathsf{f}} \in \tilde{\mathcal{G}}$ (corresponding to flow node f in the original graph $\mathcal{G}$) under Maxwell decoding on the equivalent graph $\tilde{\mathcal{G}}$ is denoted by $\hat{\phi}_{\tilde{\mathcal{G}}}^{(\ell),\mathrm{M}}(\tilde{\mathsf{f}})$. Its label is denoted by $\mathsf{l}_{\tilde{\mathcal{G}}}^{(\ell),\mathrm{M}}(\tilde{\mathsf{f}})$. Based on the above discussion, the proposition below (which is analogous to Proposition 1 for the MP decoder) follows by inspection.

Proposition 2. *For any given CB graph $\mathcal{G}$ and iteration number $\ell \geq 1$ (with respect to the original graph $\mathcal{G}$),*

$$\mathsf{l}_{\mathcal{G}}^{(2\ell),\mathrm{M}}(\mathsf{f}) = \mathsf{l}_{\tilde{\mathcal{G}}}^{(\ell),\mathrm{M}}(\tilde{\mathsf{f}}),\ \forall \mathsf{f} \in \mathcal{G},$$

and for all $\mathsf{f} \in \mathcal{G}$ that are labeled as "known",

$$\hat{\phi}_{\mathcal{G}}^{(2\ell),\mathrm{M}}(\mathsf{f}) = \hat{\phi}_{\tilde{\mathcal{G}}}^{(\ell),\mathrm{M}}(\tilde{\mathsf{f}}),$$

i.e., the two graphs are iteration-by-iteration *equivalent.*

We have the following result.

Theorem 3. *The area threshold from Definition 2, $\bar{\epsilon}$, is an upper bound on the Maxwell decoding threshold.*

Proof: See Appendix B. □

Remark 1. *The proof of Theorem 3 in Appendix B is based on DE of the Maxwell decoder as outlined above. As in Section IV, two consecutive iterations are combined into a single one, i.e., we perform the analysis on the equivalent graph of Section III. From Proposition 1, Proposition 2, and the equivalence of the DE on the two graphs, it follows directly that the upper bound from Theorem 3 applies also to the original CB graph.*

Below, we provide an analogue to [26, Th. 3.121] for the special case of $\epsilon = \bar{\epsilon}$ for the first statement of the theorem. The theorem below, together with Theorem 3 and the simulation results from Section VII of the Maxwell decoder, lead to the conjecture that the area threshold is in fact equal to the Maxwell decoding threshold.

Theorem 4. *The EMP EXIT curve (defined for the equivalent graph as in (15)) for the expected residual CB graph when the peeling decoder stops is given in parametric form by*

$$(\tilde{\epsilon}, \tilde{h}^{\mathrm{EMP}}) = (\tilde{\epsilon}(z;x), (1-\tilde{\rho}(1-z;x))^{k}),$$

where

$$\tilde{\epsilon}(z;x) = \frac{z}{(1-\tilde{\rho}(1-z;x))^{k-1}} \text{ and } \tilde{\rho}(z;x) = 1 - \frac{g(x-zx)}{g(x)},$$

and where $x = x(\epsilon)$ is the largest fixed-point of the DE recursion in (14) for a given ϵ. Furthermore, for $\epsilon = \bar{\epsilon}$ (the area threshold from Definition 2), the area $\int_0^1 \tilde{h}^{\mathrm{EMP}}(z;x)\, \mathrm{d}\tilde{\epsilon}(z;x)$ is equal to zero.

Proof: See Appendix C. □

Conjecture 1. *The area threshold from Definition 2, $\bar{\epsilon}$, is equal to the Maxwell decoding threshold and thus a lower bound on the MAP decoding threshold.*

Note that ML decoding of CBs is equivalent to solving a system of linear equations (with binary coefficient matrix) over the set of positive integers larger than or equal to $f_{\min}$. This problem resembles the well-known *integer knapsack problem*, which is known to be NP-hard. It can be shown that having a polynomial-time algorithm for the decoding of CBs implies having a polynomial-time algorithm for the integer knapsack problem. Thus, ML decoding of CBs is an NP-hard problem. This is in contrast to ML decoding of LDPC codes on the BEC which can be done in polynomial time, since it is equivalent to solving a system of linear equations over the binary field. Note that the ML decoding problem (which is equivalent in terms of error performance to the Maxwell decoding problem) of CBs can be cast as a linear integer program as follows,

$$\begin{array}{rl} \text{maximize} & \sum_{\mathsf{f} \in \mathcal{F}} \phi(\mathsf{f}) \\ \text{s.t.} & \sum_{\mathsf{f} \in \Gamma(\mathsf{c})} \phi(\mathsf{f}) = \phi(\mathsf{c}),\ \forall \mathsf{c} \in \mathcal{C} \text{ and} \\ & \phi(\mathsf{f}) \geq f_{\min},\ \forall \mathsf{f} \in \mathcal{F} \end{array} . \tag{21}$$

Thus, when simulating the Maxwell decoder in Section VII, we do not implement it using the guessing framework above (which is very useful for asymptotic analysis using DE as shown in Appendix B), but instead solve the linear integer program in (21) using the commercial Gurobi software [27].

VI. Spatially-Coupled Counter Braids

We consider the ensemble (λ, ρ, N, w) of single-layer SC-CBs (coupling the original bipartite graph), where N is the coupling chain length and w, $1 \leq w \leq N+1$, is a smoothing parameter [7]. The corresponding ensemble of SC graphs is denoted by $\mathcal{G}_{\mathrm{c}}(\lambda, \rho, N, w)$. The ensemble is constructed as follows. A collection of N flow-node groups are placed at positions $\{1, 2, \ldots, N\}$, each containing κ nodes of degree k, such that $\kappa = \frac{m_0}{N}$ (the total number of flow nodes, i.e., the number of distinct flows to be counted, is m_0). Furthermore, a collection of $M = N + w - 1$ counter-node groups are placed at positions $\{1, 2, \ldots, M\}$, each containing $\frac{\kappa R_j L'(1)}{R'(1)} = \frac{\kappa R_j k}{\gamma}$ nodes of degree j (we implicitly assume that N is chosen such that $\frac{\kappa R_j k}{\gamma}$ and also $\frac{m_0}{N}$ are integers).

The κk edge sockets in each group of flow and counter nodes are partitioned into w equally-sized subgroups (assuming that $\frac{\kappa k}{w}$ is an integer) using a uniform random interleaver. We denote by $\mathcal{P}_{n,i}^{(\mathsf{f})}$ and $\mathcal{P}_{n,i}^{(\mathsf{c})}$ the set of flow and counter node sockets, respectively, in the i-th subgroup, $i = 0, 1, \ldots, w-1$, at position n, where $n = 1, \ldots, N$ for flow node sockets and $n = 1, \ldots, M$ for counter node sockets. The SC ensemble is constructed by adding edges that connect the sockets in $\mathcal{P}_{n,i}^{(\mathsf{f})}$ to the sockets in $\mathcal{P}_{n+i,i}^{(\mathsf{c})}$. Different graphs are obtained by different socket associations. Note that this construction leaves some sockets of the counter-node groups at the boundary unconnected and these are removed. In the following, we will

$$\xi_i(k,\gamma,N,w,\boldsymbol{x}^{(2\ell-2)}) = \sum_{g=1}^{N} A_{g,i}\left[\sum_{h=1}^{M} A_{g,h}\left[1-\rho\left(1-\sum_{p=1}^{N} A_{p,h}\left\{\sum_{q=1}^{M} A_{p,q}\left[1-\rho(1-x_q^{(2\ell-2)})\right]\right\}^{k-1}\right)\right]\right]^{k-1}. \tag{22}$$

$$\tilde{\xi}_i(k,\gamma,N,w,\boldsymbol{x}^{(2\ell-2)}) = \sum_{g=1}^{N} A_{g,i}\left[\sum_{h=1}^{M} A_{g,h}\left[1-\rho\left(1-\sum_{p=1}^{N} A_{p,h}\left\{\sum_{q=1}^{M} A_{p,q}\left[1-\rho(1-x_q^{(2\ell-2)})\right]\right\}^{k-1}\right)\right]\right]^{k}. \tag{23}$$

denote the coupled ensemble $\mathcal{G}_{\text{c}}(\lambda,\rho,N,w)$ by the alternative notation $\mathcal{G}_{\text{c}}(k,\gamma,N,w)$.

A. Density Evolution Recursion

Denote by $x_i^{(\ell)}$, $i=1,\dots,M$, the error probability of an outgoing message from a flow node at position i at iteration ℓ. Note that since there are no flow-node groups at positions $i>N$, initially $x_i^{(0)}=0$ for $N<i\leq M$. Furthermore, let $\boldsymbol{x}^{(\ell)}=(x_1^{(\ell)},\dots,x_M^{(\ell)})$. Using the ensemble defined above, we get the recursion $x_i^{(2\ell)}=\epsilon\cdot\xi_i(k,\gamma,N,w,\boldsymbol{x}^{(2\ell-2)})$ where the function $\xi_i(k,\gamma,N,w,\boldsymbol{x}^{(2\ell-2)})$ is given in (22) at the top of the page, and where $\boldsymbol{A}=\{A_{p,q}\}$ is the $N\times M$ matrix defined by

$$A_{p,q}=[\boldsymbol{A}]_{p,q}=\begin{cases}\frac{1}{w}, & \text{if } 1\leq q-p+1\leq w\\ 0, & \text{otherwise}\end{cases}.$$

Note that, contrary to SC-LDPC codes, for which the SC recursion contains two summations (an outer summation over N terms and an inner summation over M terms), the recursion for SC-CBs contains four summations[3] (two summations over N terms and two summations over M terms), since the update rule for the flow nodes is different for odd and even iterations. As a result, SC-CBs do not fit within the general framework of coupled scalar recursions outlined in [19].

The MP decoding threshold of the coupled ensemble $\mathcal{G}_{\text{c}}(k,\gamma,N,w)$ is defined (analogous to the uncoupled case) as $\epsilon^{\text{c}}_{\text{MP}}=\epsilon^{\text{c}}_{\text{MP}}(k,\gamma,N,w)$, where

$$\epsilon^{\text{c}}_{\text{MP}}(k,\gamma,N,w)\triangleq\sup\left\{\epsilon\in\mathcal{E}\mid\boldsymbol{x}^{(\infty)}(k,\gamma,\epsilon,N,w)=\boldsymbol{0}\right\}.$$

Alternatively, we can analyze the asymptotic behavior of SC-CBs by fixing ϵ and finding the minimum value of β, denoted by $\beta^{\text{c}}_{\text{MP}}=\beta^{\text{c}}_{\text{MP}}(k,\epsilon,N,w)$, such that decoding is successful, where

$$\beta^{\text{c}}_{\text{MP}}(k,\epsilon,N,w)\triangleq\inf\left\{\beta>0\mid\boldsymbol{x}^{(\infty)}(k,k/\beta,\epsilon,N,w)=\boldsymbol{0}\right\}.$$

Lemma 4. *The design rate $\mathcal{R}^{\text{c}}(k,\gamma,N,w,d)$ (in bits per flow) of the coupled ensemble $\mathcal{G}_{\text{c}}(k,\gamma,N,w)$ with $w\leq N+1$ is*

$$\mathcal{R}^{\text{c}}(k,\gamma,N,w,d) = \frac{dk\sum_{i=0}^{\infty}\frac{\text{e}^{-\gamma}\gamma^i}{i!}\left[N-w+1+2\sum_{j=1}^{w-1}\left(1-\left(\frac{j}{w}\right)^{i^+}\right)\right]}{\gamma N},$$

where $i^+=i$ for $i>0$, $i^+=1$ for $i=0$, and d is the depth (in number of bits) of the counters.

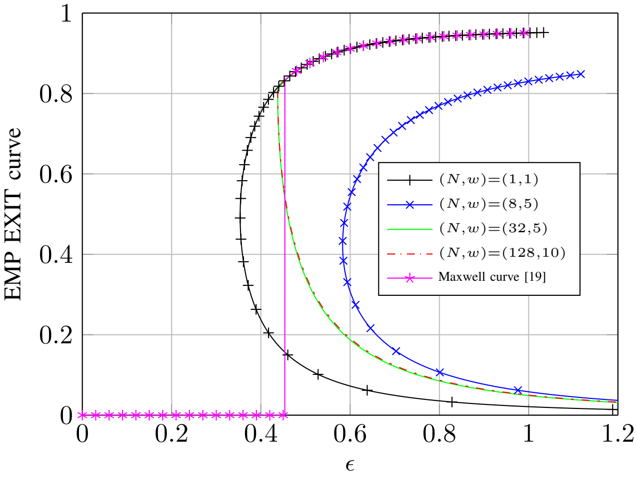

Fig. 3. EMP EXIT curves for CBs and SC-CBs with $k=3$ and $\gamma=4.233585$. For the coupled ensembles, the EMP EXIT curve of the midpoint $i=\left\lfloor\frac{M}{2}\right\rfloor$ is given.

Proof: The proof is along the same lines as the proof of [7, Lem. 3] for the design rate of regular SC-LDPC code ensembles. Details are omitted for brevity. □

For the uncoupled ensemble $\mathcal{G}_{\text{c}}(k,\gamma,1,1)$, the design rate reduces to $\frac{dk}{\gamma}=d\beta$. Not surprisingly, the coupling mechanism yields a rate increase that vanishes in the limit $N\to\infty$ (for w fixed).

Finally, we define

$$\beta^{\text{c}}=\beta^{\text{c}}(k,\gamma,N,w)\triangleq\frac{R^{\text{c}}(k,\gamma,N,w,d)}{d} \tag{24}$$

for a coupled system (as in the uncoupled case) to be the fraction of the number of counter and flow nodes.

B. Extended Message Passing Extrinsic Information Transfer Curve

As for the uncoupled system, we can define EMP EXIT curves $(\epsilon(\boldsymbol{x}),h_i^{\text{EMP}}(\boldsymbol{x}))$ for each position i, $i=1,\dots,M$, for a SC system based on fixed-points of the coupled DE recursion $x_i^{(2\ell)}=\epsilon\cdot\xi_i(k,\gamma,N,w,\boldsymbol{x}^{(2\ell-2)})$, where $\xi_i(k,\gamma,N,w,\boldsymbol{x}^{(2\ell-2)})$ is given in (22), and where $\epsilon(\boldsymbol{x})=\frac{x_i}{\xi_i(k,\gamma,N,w,\boldsymbol{x})}$, $h_i^{\text{EMP}}(\boldsymbol{x})=\tilde{\xi}_i(k,\gamma,N,w,\boldsymbol{x})$ (defined in (23) at the top of the page), and $\boldsymbol{x}$ is a fixed-point of the coupled recursion.

As an example, in Fig. 3 we plot the EMP EXIT curve $h^{\text{EMP}}_{\lfloor\frac{M}{2}\rfloor}$ (as a function of ϵ) for $k=3$ and $\gamma=4.233585$ for the pairs $(N,w)=(8,5)$, $(32,5)$, and $(128,10)$. As we can clearly

[3] Due to the band-structure of the matrix $\boldsymbol{A}$, there are only w nonzero terms in all four summations.

see from the figure, SC-CBs exhibit improved MP decoding thresholds. Furthermore, the thresholds saturate to some fixed value when N and w become large. In fact, the curves for the $(k,\gamma,32,5)$ SC-CB (the green curve) and the $(k,\gamma,128,10)$ SC-CB (the red curve) are almost indistinguishable. We also plot the *Maxwell curve* from the potential function framework of [19] (i.e., $h^{\mathrm{EMP}}(\bar{x}^*(\epsilon))$ versus ϵ where $\bar{x}^*(\epsilon)$ is taken from [19, Def. 28]). Interestingly, there is a remaining gap between $\bar{\epsilon}$ (from Definition 2) and the MP decoding thresholds of SC-CBs when N and w become large. This gap is discussed in Section VIII below.

C. Extension to More Layers

We remark that SC-CBs with more than one layer can be easily constructed by applying the spatial coupling procedure described above to each layer. As for multilayer CBs, the decoding of multilayer SC-CBs is performed layer-by-layer (see Section II-A), from the inner-most layer to the outer-most layer. Furthermore, there are no iterations between layers. Decoding is successful if the decoding of each layer (starting from the inner-most layer) is successful. Accordingly, the corresponding DE is also performed layer-by-layer.

Denote by $x_{i,l}^{(\ell)}$ the error probability of an outgoing message from a flow node at spatial position i at layer l at the ℓ-th iteration, and let $\boldsymbol{x}_l^{(\ell)} = (x_{1,l}^{(\ell)},\dots,x_{M,l}^{(\ell)})$. Here, we assume that the coupling chain length N and the smoothing parameter w are the same for all layers. Also, denote by ϵ_l the probability of observing a flow at layer l of size strictly larger than $f_{\min,l}$ ($\epsilon_1=\epsilon$). For layer one, $f_{\min,1}=f_{\min}$ (see Section II-A). On the other hand, for a layer $l>1$, $f_{\min,l}=1$, which corresponds to the case when a counter at layer $l-1$ overflows only once (the probability of this event is nonzero). Furthermore, let $\epsilon_{\mathrm{MP}}^{\mathrm{c},l}$ be the coupled MP decoding threshold of layer l, i.e.,

$$\begin{aligned}\epsilon_{\mathrm{MP}}^{\mathrm{c},l} &= \epsilon_{\mathrm{MP}}^{\mathrm{c},l}(k_l,\gamma_l,N,w)\\ &\triangleq \sup\left\{\epsilon_l\in\mathcal{E}\mid \boldsymbol{x}_l^{(\infty)}(k_l,\gamma_l,\epsilon_l,N,w)=\mathbf{0}\right\}.\end{aligned}$$

It is important to note that the input flow size distribution p_Φ will induce a certain flow size distribution at layer 2, and subsequently at the next layers. We denote by $p_{\Phi,l}^{\mathrm{i}}$ the flow size distribution at layer l induced by p_Φ. Asymptotically, for each layer, we can find the (induced) probability ϵ_l^{i} of observing a flow size strictly larger than $f_{\min,l}$ from $p_{\Phi,l}^{\mathrm{i}}$. To show the dependency of ϵ_l^{i} on p_Φ we will write $\epsilon_l^{\mathrm{i}}(p_\Phi)$. The MP decoding threshold of the multilayer SC-CB can then be computed in two steps.

1) For each layer l, compute the corresponding MP decoding threshold $\epsilon_{\mathrm{MP}}^{\mathrm{c},l}$.
2) The MP decoding threshold of the multilayer SC-CB is then

$$\begin{aligned}\epsilon_{\mathrm{MP}}^{\mathrm{c}} &= \epsilon_{\mathrm{MP}}^{\mathrm{c}}(\boldsymbol{k},\boldsymbol{\gamma},N,w)\\ &\triangleq \sup\Big\{\epsilon\in\mathcal{E}\mid \boldsymbol{x}_1^{(\infty)}(k_1,\gamma_1,\epsilon,N,w)=\mathbf{0},\\ &\qquad \epsilon_2^{\mathrm{i}}(p_\Phi)\le\epsilon_{\mathrm{MP}}^{\mathrm{c},2},\dots,\epsilon_L^{\mathrm{i}}(p_\Phi)\le\epsilon_{\mathrm{MP}}^{\mathrm{c},L}\Big\},\end{aligned}$$

where $\boldsymbol{\gamma}=(\gamma_1,\dots,\gamma_L)$ and $\boldsymbol{k}=(k_1,\dots,k_L)$.

For a given depth of the counters at layer l, d_l, the induced flow size distribution at layer $l+1$ can be easily computed. With some abuse of notation, for an arbitrary counter node c^l at layer l, $l=1,\dots,L$, let $\phi(\mathsf{c}^l)=\sum_{\mathsf{f}\in\Gamma(\mathsf{c}^l)}\phi(\mathsf{f})$ (i.e., $\phi(\mathsf{c}^l)$ ignores the fact that the counter node has finite depth). Also, let $\phi(\mathsf{f}^l)$ be the size of an arbitrary flow node f^l at layer l, $l=2,\dots,L$. We have

$$\Pr\left(\phi(\mathsf{f}^l)=a\right)=\Pr\left(a2^{d_{l-1}}\le\phi(\mathsf{c}^{l-1})<(a+1)2^{d_{l-1}}\right),$$

and

$$\begin{aligned}&\Pr\left(\phi(\mathsf{c}^l)=a\right)\\ &\quad=\sum_{b=1}^{\left\lfloor\frac{a}{f_{\min,l}}\right\rfloor}\Pr\left(\phi(\mathsf{c}^l)=a\mid \deg(\mathsf{c}^l)=b\right)\Pr\left(\deg(\mathsf{c}^l)=b\right),\end{aligned}$$

where $\deg(\mathsf{c})=|\Gamma(\mathsf{c})|$ is the degree of counter node c. Note that $\Pr\left(\deg(\mathsf{c}^l)=b\right)$ follows directly from the counter node degree distribution.

To compute $\Pr\left(\phi(\mathsf{c}^l)=a\mid \deg(\mathsf{c}^l)=b\right)$ we need to enumerate all order-b integer partitions of a (i.e., all different ways of writing a as a sum of b positive integers not considering the order of the summands) in which each summand is larger than or equal to $f_{\min,l}$. Let $\mathcal{A}(a,b)$ be the set of such integer partitions of a. Then,

$$\Pr\left(\phi(\mathsf{c}^l)=a\mid \deg(\mathsf{c}^l)=b\right)=\sum_{A\in\mathcal{A}(a,b)}\Pr(A),$$

where the probability of partition $A\in\mathcal{A}$, $\Pr(A)$, follows directly from the induced flow node degree distribution $p_{\Phi,l}^{\mathrm{i}}$ at layer l. The induced probability ϵ_l^{i} is obtained as

$$\begin{aligned}\epsilon_l^{\mathrm{i}} &= 1-\Pr\left(\phi(\mathsf{f}^l)=0\right)-\Pr\left(\phi(\mathsf{f}^l)=1\right)\\ &=1-\Pr\left(0\le\phi(\mathsf{c}^{l-1})<2^{d_{l-1}}\right)\\ &\qquad-\Pr\left(2^{d_{l-1}}\le\phi(\mathsf{c}^{l-1})<2^{d_{l-1}+1}\right)\\ &=1-\Pr\left(0\le\phi(\mathsf{c}^{l-1})<2^{d_{l-1}+1}\right).\end{aligned}$$

For the particular case of two layers, the induced probability ϵ_2^{i} is easily obtained, since only the two first terms of the first induced probability distribution $p_{\Phi,2}^{\mathrm{i}}$ are needed.

Finally, we remark that the procedure above to derive the induced flow size distributions is general and applies also to multilayer CBs (by setting the coupling chain length to $N=1$ and the smoothing parameter to $w=1$).

VII. Numerical Results

In this section, we present several numerical results that show threshold improvement, but lack of threshold saturation. In addition, we present simulation results for both the MP and the Maxwell decoder of CBs that show that, in fact, there is no threshold saturation.

We assume the power law $\Pr(\phi(\mathsf{f})>\eta)=\eta^{-\alpha}$, where η is a positive integer, for the flow size distribution [28]. Typically, flow size distributions from real Internet traces have $\alpha\approx 2$, while for distributions with a heavier tail a smaller value of α should be used [28]. Due to the nature of the power law distribution, $f_{\min}=2$ for all exponents α. The particular value of ϵ is related to α, since $\epsilon=2^{-\alpha}$. As an example, $\alpha=1.5$ gives $\epsilon=2^{-1.5}$. We use this value for ϵ in the simulations below.

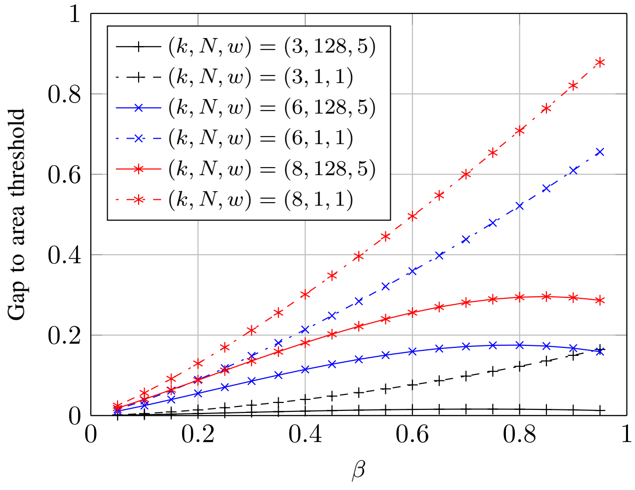

Fig. 4. The difference $\bar{\epsilon} - \epsilon^{\mathrm{c}}_{\mathrm{MP}}$, where $\bar{\epsilon}$ is the area threshold from Definition 2 and $\epsilon^{\mathrm{c}}_{\mathrm{MP}}$ is the MP decoding threshold of SC-CBs, as a function of β for different values of the left-degree k with $(N, w) = (128, 5)$ and $(1, 1)$ (uncoupled).

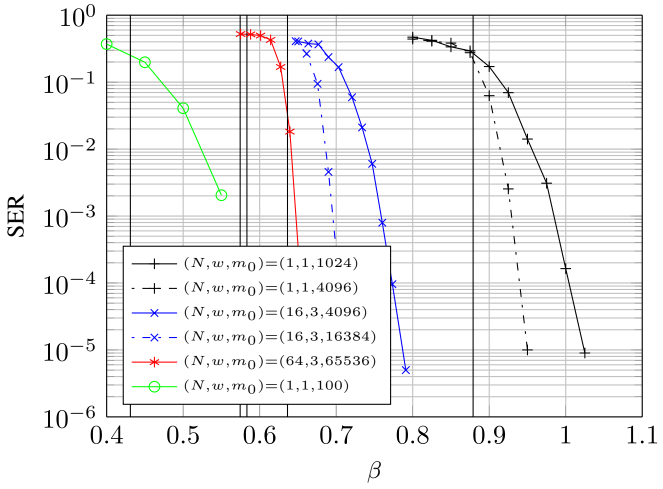

Fig. 5. SER performance of CBs and SC-CBs as a function of β for $k = 6$ and $\epsilon = 2^{-1.5}$.

A. Comparison of Area and Message Passing Decoding Thresholds

In Fig. 4, we plot the difference between the area threshold from Definition 2 (conjectured to be a lower bound on the MAP decoding threshold, see Conjecture 1) and the MP decoding threshold of SC-CBs, $\bar{\epsilon} - \epsilon^{\mathrm{c}}_{\mathrm{MP}}$, as a function of $\beta = \frac{k}{\gamma}$ for different values of the left-degree k. The SC-CBs have parameters $(N, w) = (128, 5)$. For comparison purposes, we also plot $\bar{\epsilon} - \epsilon_{\mathrm{MP}}$ for the uncoupled CBs ($(N, w) = (1, 1)$). As we can observe from the figure, there is a gap between the MP decoding threshold of SC-CBs and the conjectured lower bound on the MAP decoding threshold. Thus, surprisingly, threshold saturation does not occur for SC-CBs. However, the gap to the conjectured lower bound is significantly larger for uncoupled CBs, meaning that spatial coupling indeed improves performance. Note that the gap varies with β and depends on k.

We remark that when β becomes large (in the sense of approaching one), the area threshold $\bar{\epsilon}$ becomes larger than one. The minimum value of β for which $\bar{\epsilon}$ is larger than one, referred to as the transition point, depends on k (larger k implies a lower transition point). From the perspective of the DE this is not really a problem (the recursion is a valid recursion also for values of ϵ above one), but of course there is no physical system corresponding to such values of ϵ (since ϵ is a probability). From a practical perspective, however, there is no need to design a system with a value of β above its transition point.

B. Simulation Results

In Fig. 5, we give symbol error rate (SER) results for CBs decoded using the MP decoding algorithm and the Maxwell decoder, for $k = 6$ and $\epsilon = 2^{-1.5}$, as a function of β. In the figure we also plot the SER for several SC-CBs with parameters $(N, w, m_0) = (16, 3, 4096)$, $(16, 3, 16384)$, and $(64, 3, 65536)$, for the same k and ϵ. The Maxwell decoder has been implemented as described at the end of Section V, and due to its high computational complexity (the problem is indeed NP-hard) only a short block length ($m_0 = 100$) has been considered (the green curve in Fig. 5). For MP decoding of CBs, we have considered the block lengths $m_0 = 1024$ and 4096. The four black vertical lines show different thresholds as follows (in the order from left to right): The conjectured Maxwell decoding threshold (or the area threshold) and the coupled MP decoding thresholds $\beta^{\mathrm{c}}_{\mathrm{MP}}$ for $w = 3$ and $N = 128$, 64, and 16.

For both CBs and SC-CBs, we can clearly see how the performance improves as the block length increases (for a given coupling chain length and smoothing parameter) and approaches the corresponding MP decoding threshold. As expected, there is a larger gap to the MP decoding threshold for SC-CBs than for uncoupled CBs for a given fixed overall block length due to the fact that the block length at each position becomes shorter as the coupling chain length increases. SC-CBs achieve significantly better performance than uncoupled CBs. In particular, the threshold is improved from $\beta_{\mathrm{MP}} = 0.88$ to $\beta^{\mathrm{c}}_{\mathrm{MP}} = 0.57$ for coupling length $N = 128$ and $w = 3$. However, it is apparent from the figure that the MP decoding threshold for the coupled ensembles does not saturate to the area threshold (it remains a gap to the area threshold $\bar{\epsilon} = 0.43$). The lack of threshold saturation is also supported by the simulation results of the Maxwell decoder (green curve). Despite of the short block length, the Maxwell decoder for the uncoupled CB performs significantly better than MP decoding of SC-CBs.

C. Rate Gap to Entropy Lower Bound

In Fig. 6, we depict the rate gap of an optimized (over γ) single-layer SC-CB to the entropy lower bound of the flow size distribution (in bits per flow) as a function of α for different values of the flow node degree k. The number of spatial positions is fixed to $N = 128$ and the smoothing parameter is fixed to $w = 5$, while α is sampled from 0.5 to 2.5 in steps of 0.2. The depth d of the counters has been chosen such that the average asymptotic probability that a counter

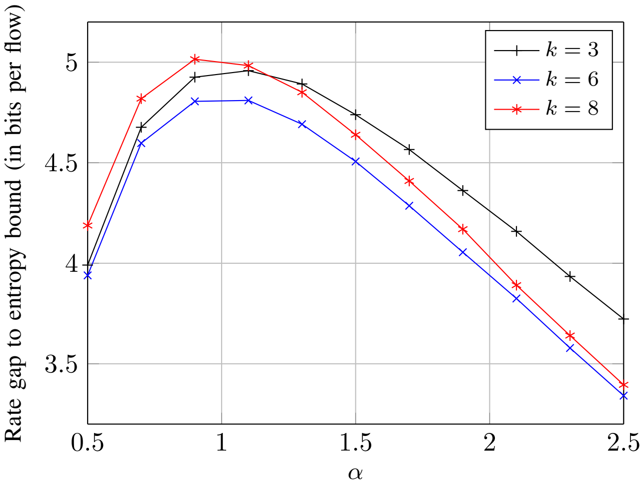

Fig. 6. Rate gap of an optimized (over γ) single-layer SC-CB to the entropy lower bound of the flow size distribution (in bits per flow) as a function of α for different values of the flow node degree k. The number of spatial positions is fixed to $N=128$ and the smoothing parameter is fixed to $w=5$.

node overflows (computed from simulations for a large value of m_0) over the ensemble of possible CBs (for given values of k, γ, and α) is at most 10^{-4}. Degree $k=6$ gives the lowest gap for all values of α. Moreover, $k=8$ gives a larger gap than $k=3$ when α is small. The reason is that a larger depth d is required to maintain the same average asymptotic overflow probability. The same observation was made in [29] for $\alpha=1.5$.

For $\alpha=1.5$, the optimized SC-CB with $(k,N,w)=(6,128,5)$ has a gap to the entropy lower bound of approximately 4.5 bits per flow. For an uncoupled CB with $k=3$ (which gives the best performance, since the performance degrades with increasing k as shown in [29, Fig. 3]) the corresponding rate gap (with the same overflow probability) is approximately 5.3 bits per flow,[4] meaning that spatial coupling gives a gain of approximately 0.8 bits per flow.

The rather large gap to the entropy lower bound in Fig. 6 is due to having only a single-layer system. To increase performance, more layers are needed. As shown in Figs. 4, 5, and 6, spatial coupling significantly improves performance of single-layer uncoupled CBs. Since the decoding of multilayer CBs proceeds layer-by-layer, we also expect multilayer SC-CBs to significantly improve the performance of their uncoupled counterparts: Spatial coupling will improve the decoding threshold for each layer and hence a better overall decoding threshold (see Section VI-C) is expected.

VIII. Discussion on the Gap to the Area Threshold

As shown in the previous section, we observe an improvement of the MP decoding threshold with spatial coupling, but no threshold saturation to the potential threshold by coupling the original bipartite graph (see Section VI). The gap to the area threshold seems to be fundamental and is still present even for other more structured ways to do spatial coupling. For instance, the gap is still present with regular right degree distributions, i.e., for SC-CBs where the underlying uncoupled ensemble has a regular right degree distribution, as can be seen in Fig. 7. Different curves correspond to different values of β. For instance, for the two blue curves (the dash-dotted curve is for uncoupled CBs, while the solid curve is for SC-CBs), the left-degree k increases from 3 to 8, while the right-degree is equal to $2k$, which results in $\beta=1/2$. Again, we observe a gap between the MP decoding threshold of $(k,\gamma,128,5)$ SC-CBs and the area threshold. Note that the gap grows with k and depends on β, as also shown in Fig. 4. Also, as shown in Fig. 4, the gap to area threshold is significantly larger for uncoupled CBs, meaning that spatial coupling indeed improves performance for regular CBs.

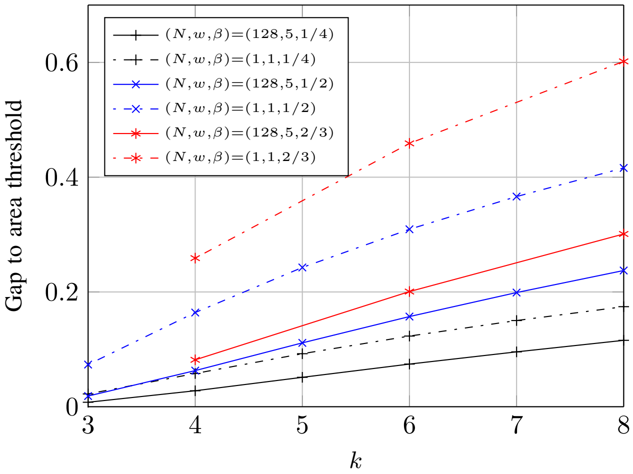

Fig. 7. The difference $\bar{\epsilon}-\epsilon^{\mathrm{c}}_{\mathrm{MP}}$, where $\bar{\epsilon}$ is the area threshold from Definition 2 and $\epsilon^{\mathrm{c}}_{\mathrm{MP}}$ is the MP decoding threshold of *regular* SC-CBs, as a function of the left-degree k for different values of β with $(N,w)=(128,5)$ and $(1,1)$ (uncoupled).

We remark that coupling the equivalent graph representation of Section III gives exactly the same coupled DE recursion $x_i^{(2\ell)}=\epsilon\cdot\xi_i(k,\gamma,N,w,\boldsymbol{x}^{(2\ell-2)})$, where $\xi_i(k,\gamma,N,w,\boldsymbol{x}^{(2\ell-2)})$ is given in (22). However, interestingly, if we consider the *coupled version* of (11)–(13), i.e., we substitute y and x in (12) and (13), respectively, by an average over spatial positions, we do indeed observe threshold saturation to the potential threshold. The DE recursion derived below corresponds to the coupled version of (11)–(13), split into two steps.

Let $x_i^{(\ell)}$ and $y_i^{(\ell)}$, $i=1,\dots,M$, denote the output message error probability at a flow and counter node, respectively, at the ℓ-th iteration at coupling chain position i. As before, we initialize $x_i^{(0)}=0$ for $N<i\leq M$. Now, define the following

[4] The gap of approximately 5.3 bits per flow to the entropy lower bound matches well with the results in [1, Fig. 12]. Note that in [1, Fig. 12] the entropy lower bound is shown to be slightly below 3 bits per flow. However, this seems to be a misprint, as it is just below 2 bits per flow for $\alpha=1.5$. Moreover, the explicit overflow probability used when calculating the rate is not explicitly stated in [1].

DE equations at iteration ℓ,

$$y_i^{(\ell)} = \begin{cases} 1-\rho\left(1-\frac{1}{w}\sum_{j=0}^{\min(i-1,w-1)} x_{i-j}^{(\ell-1)}\right), & \text{if } \ell \text{ is odd} \\ 1-\rho\left(1-x_i^{(\ell-1)}\right), & \text{if } \ell \text{ is even} \end{cases},$$

$$x_i^{(\ell)} = \begin{cases} \left(y_i^{(\ell)}\right)^{k-1}, & \text{if } \ell \text{ is odd} \\ \epsilon\left(\frac{1}{w}\sum_{j=0}^{\min(M-i,w-1)} y_{i+j}^{(\ell)}\right)^{k-1}, & \text{if } \ell \text{ is even} \end{cases}.$$

With this modified DE, threshold saturation to the potential threshold, which (from Conjecture 1) is a lower bound on the MAP decoding threshold, can be proved using the potential function framework by Yedla *et al.* outlined in [19]. This DE is characterized by an average only for odd and even iterations for the counter node and flow node updates, respectively. However, when coupling the original (or the equivalent) graph in the standard way, the average appears for all iterations (hence the four summations in (22)). This effect, which is due to the fact that, as opposed to LDPC codes, the flow node update is different for odd and even iterations, seems to be the responsible for the lack of threshold saturation.

A question that remains open is whether the DE equations above correspond to a physical system, i.e., whether threshold saturation can be achieved with an alternative coupling.

The lack of threshold saturation has been observed in the past for other coupled systems. For example, SC-LDPC codes decoded using the corrected min-sum decoding algorithm show no threshold saturation, albeit some improvement to the threshold [30]. The lack of threshold saturation has also been observed from simulation of linear programming decoding of long SC codes using the alternating direction method of multipliers [30].

It is interesting to remark, however, that threshold saturation has been numerically observed in [10] for iterative HDD of SC generalized LDPC codes even with sub-optimal (bounded distance) component decoding, which shows that the lack of saturation cannot by itself be explained by the hard-decision nature of the MP decoding algorithm. In fact, a soft-decision SP decoding algorithm as outlined in (2)–(3) could possibly perform better than the MP decoding algorithm studied in this paper. However, the complexity of such an algorithm is significantly higher.

IX. Connection With Compressed Sensing

In this section, we explore the connection of CBs to CS as established in [2].

We consider a single-layer SC-CB with m_0 flow nodes, $\mathsf{f}_1, \dots, \mathsf{f}_{m_0}$, corresponding to the distinct flows whose sizes are to be measured, and m_1 counter nodes. We define the flow size vector $\boldsymbol{\phi} = (\phi_1, \dots, \phi_{m_0})$, of length m_0, where $\phi_i = \phi(\mathsf{f}_i)$ is the actual flow size of flow i. As in Section IV, the minimum flow size is $f_{\min}$, i.e., $\phi_i \geq f_{\min}\ \forall i$. Now, if $f_{\min} = 0$, the vector $\boldsymbol{\phi}$ has a number of zero entries and the rest are *nonzero* positive integers. Thus, $\boldsymbol{\phi}$ can be interpreted as a *nonnegative* (integer) signal vector. In this case, the m_1 counters can be also regarded as m_1 linear measurements of $\boldsymbol{\phi}$. If the number of nonzero entries in $\boldsymbol{\phi}$, denoted by t, is small (as compared to m_0), $\boldsymbol{\phi}$ is a t-sparse vector. Thus, the CB architecture can be regarded as a CS scheme with the important difference that for CBs, the signal is nonnegative. The MP decoding of the CB corresponds now to the recovery of the vector $\boldsymbol{\phi}$ based on the m_1 linear measurements. We remark that, while for CBs $\boldsymbol{\phi}$ is a vector with nonnegative integer entries, the asymptotic analysis in Section IV depends only on the properties of the CB graph and ϵ, and not on the exact value of $f_{\min}$ or the fact that the flow sizes are integers. Thus, our previous analysis is also valid for CS of nonnegative real signals.

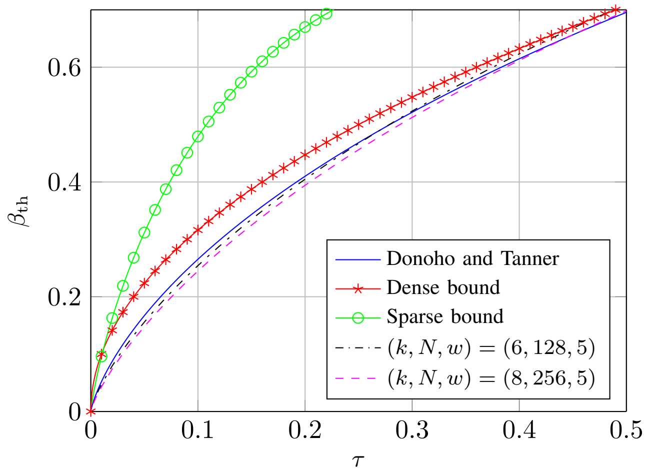

Fig. 8. Undersampling-sparsity phase transition trajectory (under MP decoding) of SC-CBs compared to two bounds from [2] and the Donoho-Tanner phase transition trajectory for random Gaussian measurement matrices with ℓ_1-norm minimization reconstruction.

Let $\mathbb{R}_{\geq 0}$ denote the set of nonnegative real numbers. Given a vector $\boldsymbol{\phi} \in \mathbb{R}_{\geq 0}^{m_0}$, of size m_0, with t nonzero entries, we define the *sparsity ratio* $\tau = \frac{t}{m_0}$ as the probability that a given entry of the signal vector $\boldsymbol{\phi}$ is nonzero. Note that $\tau = \epsilon$, i.e., the probability of observing a flow size strictly larger than $f_{\min} = 0$ (see Section IV). Furthermore, we define the *undersampling ratio* (or factor) as the number of measurements over the signal dimension, m_0. This corresponds precisely to the parameter $\beta = m_1/m_0$ defined in Section IV for uncoupled CBs. For SC-CBs the undersampling ratio is equal to $\beta^{\rm c}$ (taking into account the rate increase due to edge effects (or termination) for SC-CBs) defined in (24). We also define the undersampling threshold, $\beta_{\rm th}$, as the lowest $\beta^{\rm c}$ (for a given sparsity ratio τ) such that the error probability for recovering t-sparse vectors goes to zero as $m_0 \to \infty$ and $t \to \infty$, while the ratio $\frac{t}{m_0}$ is fixed and equal to τ.

In Fig. 8, we plot $\beta_{\rm th}$ as a function of τ for single-layer SC-CBs, with parameters $(k, N, w) = (6, 128, 5)$ and $(8, 256, 5)$. The curve $\beta_{\rm th}(\tau)$ is commonly known as the phase transition curve in the CS literature. For comparison purposes, we also report in the figure the corresponding threshold $\beta_{\rm th}$ for the two explicit constructions of CBs from [2], which we refer to as *sparse bound* and *dense bound* (see [2]), respectively. The sparse bound is for a family of left-regular CBs with $\gamma = \ln(2)/\epsilon$, $k = \ln(1/\epsilon)/\ln(2)$, and $\beta(\epsilon) = 1/(\ln 2)^2 \cdot \epsilon \cdot \ln(1/\epsilon)$ (see [2, Th. 5] for details). The dense bound is for a family of irregular CBs (found by making use of results from the optimization of LDPC code ensembles over the BEC) and has

$\beta(\epsilon)=\sqrt{\epsilon}$ (see [2, Th. 6]). It is observed that SC-CBs yield a significant improvement of β_{th}.

In Fig. 8, we also compare the phase transition curve of SC-CBs to that of the ensemble of random Gaussian measurement matrices with ℓ_1-norm minimization reconstruction by Donoho and Tanner [4]. The curve is taken from [4] (see also [31], which contains the actual data in Matlab format). Surprisingly, SC-CBs give an improved phase transition trajectory in the sparse region. As compared to the ℓ_1-norm minimization reconstruction algorithm in [4], CS with CBs has the advantage of using sparse measurement matrices, i.e., measurements can be taken in $O(1)$ operations, and of a corresponding MP decoding algorithm of low complexity (linear in the signal dimension, $O(m_0)$), where $O(\cdot)$ is the standard O-notation.

Finally, we remark that iterative decoding for CS has also been considered in the literature in [16], [21], [32]–[36]. One such algorithm is the interval-passing algorithm by Chandar *et al.* [21] for binary measurements matrices, which was further generalized to nonbinary measurements matrices in [33]. In fact, for CBs the interval-passing algorithm is equivalent to the MP decoding algorithm outlined in [1] and presented here in Section II-A.

X. Conclusion

We proposed spatial coupling of CBs as a means of improving the MP decoding performance of (uncoupled) CBs. As expected, SC-CBs yield significantly better MP decoding thresholds than uncoupled CBs. However, surprisingly, our results suggest that the well-known phenomenon of threshold saturation does not occur for SC-CBs. We therefore analyzed the asymptotic performance of single-layer CBs. In particular, we introduced an equivalent graph representation of CBs with identical asymptotic and iteration-by-iteration finite-length performance to that of the original CB graph, which makes the analysis more amenable. Based on this equivalent representation, we showed that the area threshold and the potential threshold of CBs are equal. We also derived the Maxwell decoder for CBs and formulated the conjecture that the area threshold is equal to the Maxwell decoding threshold and therefore a lower bound on the MAP decoding threshold. There is a gap between the MP decoding threshold of SC-CBs and the conjectured MAP decoding threshold, showing that threshold saturation does not occur. Simulation results of the Maxwell decoder for finite length, which show better performance than MP decoding of SC-CBs for larger block lengths, support the conjecture. Finally, we also considered SC-CBs for CS and showed that they can yield very low undersampling thresholds.

Appendix A
Derivation of the Message Passing Decoding Algorithm in [1]

We derive the MP decoding algorithm proposed in [1] starting from the flow node and counter node updates $\mu_{\mathsf{f}\to\mathsf{c}}^{\mathsf{MaxP},(\ell)}(a)$ and $\psi_{\mathsf{c}\to\mathsf{f}}^{\mathsf{MaxP},(\ell)}(a)$ of the max-product algorithm in (5) and (4), respectively. We call the *support* of a probability distribution $h(a)$ the set $\mathcal{A}$ of values for which $h(a)\neq 0$, i.e., $\mathcal{A}=\{a : h(a)\neq 0\}$. Note that for CBs a is a nonnegative integer. Furthermore, it follows that the support $\mathcal{A}_{\mu}^{(\ell)}$ of $\mu_{\mathsf{f}\to\mathsf{c}}^{\mathsf{MaxP},(\ell)}(a)$ is the set of all integers in a given range $[L_{\mathsf{f}\to\mathsf{c}}^{(\ell)},U_{\mathsf{f}\to\mathsf{c}}^{(\ell)}]$. Initially, $\mathcal{A}_{\mu}^{(\ell)}$ is the set of all integers in $[f_{\min},\infty)$ (we know that the minimum flow size is $f_{\min}$ and there is no limit in the maximum flow size). Likewise, the support $\mathcal{A}_{\psi}^{(\ell)}$ of $\psi_{\mathsf{c}\to\mathsf{f}}^{\mathsf{MaxP},(\ell)}(a)$ consists of all integers within a range $[L_{\mathsf{c}\to\mathsf{f}}^{(\ell)},U_{\mathsf{c}\to\mathsf{f}}^{(\ell)}]$. Therefore, the supports $\mathcal{A}_{\mu}^{(\ell)}$ and $\mathcal{A}_{\psi}^{(\ell)}$ are completely defined by the ranges $[L_{\mathsf{f}\to\mathsf{c}}^{(\ell)},U_{\mathsf{f}\to\mathsf{c}}^{(\ell)}]$ and $[L_{\mathsf{c}\to\mathsf{f}}^{(\ell)},U_{\mathsf{c}\to\mathsf{f}}^{(\ell)}]$, respectively, and as a consequence by the left and right limits of the ranges. Equivalently, we will write $L_{\mathsf{f}\to\mathsf{c}}^{(\ell)}\leq a\leq U_{\mathsf{f}\to\mathsf{c}}^{(\ell)}$ and $L_{\mathsf{c}\to\mathsf{f}}^{(\ell)}\leq a\leq U_{\mathsf{c}\to\mathsf{f}}^{(\ell)}$. We would like to remark that the supports $\mathcal{A}_{\mu}^{(\ell)}$ and $\mathcal{A}_{\psi}^{(\ell)}$ are iteration-dependent. In particular, the limits of the corresponding ranges change with iterations. In the following, with some abuse of language, we will refer to the range on which a support is defined as the support itself.

Assuming that the decoder does not exploit any information about the flow size distribution, the message from flow node f to counter node c at iteration $\ell=0$ can be initialized as

$$\mu_{\mathsf{f}\to\mathsf{c}}^{\mathsf{MaxP},(0)}(a)=\begin{cases}\frac{1}{\phi(\mathsf{c})-f_{\min}+1}, & \text{if } L_{\mathsf{f}\to\mathsf{c}}^{(0)}\leq a\leq U_{\mathsf{f}\to\mathsf{c}}^{(0)}\\ 0, & \text{otherwise}\end{cases},$$

where $L_{\mathsf{f}\to\mathsf{c}}^{(0)}=f_{\min}$ and $U_{\mathsf{f}\to\mathsf{c}}^{(0)}=\phi(\mathsf{c})$, i.e., the flow size distribution is assumed to be uniform over its support $[L_{\mathsf{f}\to\mathsf{c}}^{(0)},U_{\mathsf{f}\to\mathsf{c}}^{(0)}]$. For notational simplicity, we define $K_{\mathsf{c}}\triangleq\frac{1}{\phi(\mathsf{c})-f_{\min}+1}$.

Consider now the first iteration, $\ell=1$, of the max-product algorithm. The update rule (4) can now be rewritten as

$$\psi_{\mathsf{c}\to\mathsf{f}}^{\mathrm{MaxP},(1)}(a)=\begin{cases}K_{\mathsf{c}}^{|\Gamma(\mathsf{c})|-1}, & \text{if } L_{\mathsf{c}\to\mathsf{f}}^{(1)}\leq a\leq U_{\mathsf{c}\to\mathsf{f}}^{(1)}\\ 0, & \text{otherwise}\end{cases},$$

where $L_{\mathsf{c}\to\mathsf{f}}^{(1)}=\max\{\phi(\mathsf{c})-\sum_{\mathsf{f}'\in\Gamma(\mathsf{c})\backslash\mathsf{f}}U_{\mathsf{f}'\to\mathsf{c}}^{(0)},f_{\min}\}$, $U_{\mathsf{c}\to\mathsf{f}}^{(1)}=\max\{\phi(\mathsf{c})-\sum_{\mathsf{f}'\in\Gamma(\mathsf{c})\backslash\mathsf{f}}L_{\mathsf{f}'\to\mathsf{c}}^{(0)},f_{\min}\}$, and the support of the distribution $\psi_{\mathsf{c}\to\mathsf{f}}^{\mathsf{MaxP},(1)}(a)$ is $[L_{\mathsf{c}\to\mathsf{f}}^{(1)},U_{\mathsf{c}\to\mathsf{f}}^{(1)}]$. The values $L_{\mathsf{c}\to\mathsf{f}}^{(1)}$ and $U_{\mathsf{c}\to\mathsf{f}}^{(1)}$ come directly from the fact that the value of a counter node is equal to the sum of the values of its neighboring flow nodes. On the flow node side, the update rule in (5) can be written as

$$\mu_{\mathsf{f}\to\mathsf{c}}^{\mathrm{MaxP},(1)}(a)=\begin{cases}\left(K_{\mathsf{c}}^{|\Gamma(\mathsf{c})|-1}\right)^{|\Gamma(\mathsf{f})|-1}, & \text{if } L_{\mathsf{f}\to\mathsf{c}}^{(1)}\leq a\leq U_{\mathsf{f}\to\mathsf{c}}^{(1)}\\ 0, & \text{otherwise}\end{cases}$$

where $L_{\mathsf{f}\to\mathsf{c}}^{(1)}=\max_{\mathsf{c}'\in\Gamma(\mathsf{f})\backslash\mathsf{c}}L_{\mathsf{c}'\to\mathsf{f}}^{(1)}$ and $U_{\mathsf{f}\to\mathsf{c}}^{(1)}=\min_{\mathsf{c}'\in\Gamma(\mathsf{f})\backslash\mathsf{c}}U_{\mathsf{c}'\to\mathsf{f}}^{(1)}$. The new support $[L_{\mathsf{f}\to\mathsf{c}}^{(1)},U_{\mathsf{f}\to\mathsf{c}}^{(1)}]$ comes directly from the fact that the update rule of (5) is the product of the incoming messages.

Consider now the second iteration from counter nodes to flow nodes. We get

$$\psi_{\mathsf{c}\to\mathsf{f}}^{\mathrm{MaxP},(2)}(a)$$
$$=\begin{cases}\left(\left(K_{\mathsf{c}}^{|\Gamma(\mathsf{c})|-1}\right)^{|\Gamma(\mathsf{f})|-1}\right)^{|\Gamma(\mathsf{c})|-1}, & \text{if } L_{\mathsf{c}\to\mathsf{f}}^{(2)}\leq a\leq U_{\mathsf{c}\to\mathsf{f}}^{(2)}\\ 0, & \text{otherwise}\end{cases},$$

where $L^{(2)}_{\mathsf{c}\to\mathsf{f}} = \max\{\phi(\mathsf{c}) - \sum_{\mathsf{f}'\in\Gamma(\mathsf{c})\backslash\mathsf{f}} U^{(1)}_{\mathsf{f}'\to\mathsf{c}}, f_{\min}\}$ and $U^{(2)}_{\mathsf{c}\to\mathsf{f}} = \max\{\phi(\mathsf{c}) - \sum_{\mathsf{f}'\in\Gamma(\mathsf{c})\backslash\mathsf{f}} L^{(1)}_{\mathsf{f}'\to\mathsf{c}}, f_{\min}\}$. Similarly, one can derive the messages $\psi^{\mathsf{MaxP},(\ell)}_{\mathsf{c}\to\mathsf{f}}(a)$ and $\mu^{\mathsf{MaxP},(\ell)}_{\mathsf{f}\to\mathsf{c}}(a)$ for a generic iteration ℓ,

$$\psi^{\mathsf{MaxP},(\ell)}_{\mathsf{c}\to\mathsf{f}}(a) = \begin{cases} A^{(\ell)}, & \text{if } L^{(\ell)}_{\mathsf{c}\to\mathsf{f}} \le a \le U^{(\ell)}_{\mathsf{c}\to\mathsf{f}} \\ 0, & \text{otherwise} \end{cases}, \tag{25}$$

$$\mu^{\mathsf{MaxP},(\ell)}_{\mathsf{f}\to\mathsf{c}}(a) = \begin{cases} B^{(\ell)}, & \text{if } L^{(\ell)}_{\mathsf{f}\to\mathsf{c}} \le a \le U^{(\ell)}_{\mathsf{f}\to\mathsf{c}} \\ 0, & \text{otherwise} \end{cases}, \tag{26}$$

where $A^{(\ell)}$ and $B^{(\ell)}$ are constants and

$$L^{(\ell)}_{\mathsf{c}\to\mathsf{f}} = \max\left\{\phi(\mathsf{c}) - \sum_{\mathsf{f}'\in\Gamma(\mathsf{c})\backslash\mathsf{f}} U^{(\ell-1)}_{\mathsf{f}'\to\mathsf{c}}, f_{\min}\right\}, \tag{27}$$

$$U^{(\ell)}_{\mathsf{c}\to\mathsf{f}} = \max\left\{\phi(\mathsf{c}) - \sum_{\mathsf{f}'\in\Gamma(\mathsf{c})\backslash\mathsf{f}} L^{(\ell-1)}_{\mathsf{f}'\to\mathsf{c}}, f_{\min}\right\}, \tag{28}$$

$$L^{(\ell)}_{\mathsf{f}\to\mathsf{c}} = \max_{\mathsf{c}'\in\Gamma(\mathsf{f})\backslash\mathsf{c}} L^{(\ell)}_{\mathsf{c}'\to\mathsf{f}}, \tag{29}$$

$$U^{(\ell)}_{\mathsf{f}\to\mathsf{c}} = \min_{\mathsf{c}'\in\Gamma(\mathsf{f})\backslash\mathsf{c}} U^{(\ell)}_{\mathsf{c}'\to\mathsf{f}}. \tag{30}$$

It is important to observe that the messages exchanged between flow nodes and counter nodes in each iteration (see (25)–(26)) are constant within a given support and 0 otherwise. Therefore, it is enough to exchange the limits of the supports $L^{(\ell)}_{\mathsf{f}\to\mathsf{c}}, U^{(\ell)}_{\mathsf{f}\to\mathsf{c}}$ and $L^{(\ell)}_{\mathsf{c}\to\mathsf{f}}, U^{(\ell)}_{\mathsf{c}\to\mathsf{f}}$ at each iteration, and we can simplify the update rules of the max-product algorithm (25)–(26) as

$$\psi^{\mathsf{MaxP},(\ell)}_{\mathsf{c}\to\mathsf{f}} = (L^{(\ell)}_{\mathsf{c}\to\mathsf{f}}, U^{(\ell)}_{\mathsf{c}\to\mathsf{f}}), \tag{31}$$

$$\mu^{\mathsf{MaxP},(\ell)}_{\mathsf{f}\to\mathsf{c}} = (L^{(\ell)}_{\mathsf{f}\to\mathsf{c}}, U^{(\ell)}_{\mathsf{f}\to\mathsf{c}}), \tag{32}$$

i.e., each message is a two-dimensional vector. Note that these vectors convey all relevant information. Furthermore, it can be easily shown by induction that, for $\ell \ge 1$,

$$\begin{aligned} L^{(2\ell-1)}_{\mathsf{f}\to\mathsf{c}} &= L^{(2\ell-2)}_{\mathsf{f}\to\mathsf{c}}, & U^{(2\ell)}_{\mathsf{f}\to\mathsf{c}} &= U^{(2\ell-1)}_{\mathsf{f}\to\mathsf{c}}, \\ L^{(2\ell-1)}_{\mathsf{c}\to\mathsf{f}} &= L^{(2\ell-2)}_{\mathsf{c}\to\mathsf{f}}, & U^{(2\ell)}_{\mathsf{c}\to\mathsf{f}} &= U^{(2\ell-1)}_{\mathsf{c}\to\mathsf{f}}, \end{aligned}$$

where $L^{(0)}_{\mathsf{c}\to\mathsf{f}} \triangleq f_{\min}$. Thus, for odd iterations it is only necessary to exchange the upper limits $U^{(2\ell-1)}_{\mathsf{f}\to\mathsf{c}}$ and $U^{(2\ell-1)}_{\mathsf{c}\to\mathsf{f}}$. Similarly, for even iterations it is only necessary to exchange the lower limits $L^{(2\ell)}_{\mathsf{f}\to\mathsf{c}}$ and $L^{(2\ell)}_{\mathsf{c}\to\mathsf{f}}$. Therefore, we can rewrite (31) and (32) as

$$\psi^{\mathsf{MaxP},(\ell)}_{\mathsf{c}\to\mathsf{f}} = \begin{cases} U^{(\ell)}_{\mathsf{c}\to\mathsf{f}}, & \text{if } \ell \text{ is odd} \\ L^{(\ell)}_{\mathsf{c}\to\mathsf{f}}, & \text{if } \ell \text{ is even} \end{cases}, \tag{33}$$

$$\mu^{\mathsf{MaxP},(\ell)}_{\mathsf{f}\to\mathsf{c}} = \begin{cases} U^{(\ell)}_{\mathsf{f}\to\mathsf{c}}, & \text{if } \ell \text{ is odd} \\ L^{(\ell)}_{\mathsf{f}\to\mathsf{c}}, & \text{if } \ell \text{ is even} \end{cases}. \tag{34}$$

Finally, using (27)–(30) and observing that the structure of the update rules in (27) and (28) is the same, (33) and (34) can be rewritten as

$$\psi^{\mathsf{MaxP},(\ell)}_{\mathsf{c}\to\mathsf{f}} = \max\left\{\phi(\mathsf{c}) - \sum_{\mathsf{f}'\in\Gamma(\mathsf{c})\backslash\mathsf{f}} \mu^{\mathsf{MaxP},(\ell-1)}_{\mathsf{f}'\to\mathsf{c}},\ f_{\min}\right\},$$

$$\mu^{\mathsf{MaxP},(\ell)}_{\mathsf{f}\to\mathsf{c}} = \begin{cases} \min_{\mathsf{c}'\in\Gamma(\mathsf{f})\backslash\mathsf{c}} \psi^{\mathsf{MaxP},(\ell)}_{\mathsf{c}'\to\mathsf{f}}, & \text{if } \ell \text{ is odd} \\ \max_{\mathsf{c}'\in\Gamma(\mathsf{f})\backslash\mathsf{c}} \psi^{\mathsf{MaxP},(\ell)}_{\mathsf{c}'\to\mathsf{f}}, & \text{if } \ell \text{ is even} \end{cases},$$

which is the MP decoding algorithm proposed in [1].

Notice that if the summation within the counter nodes had been over some *group*, i.e., the flow node values were elements of some group, then the support of the flow size distribution would have stayed the same throughout the iterations, and thus the MP algorithm would not work at all. Also, in this case, both the SP and the max-product algorithms would have been identical. The crucial point for reducing the support of the flow size distribution throughout the iterations of the algorithm is the nonexistence of additive inverses. Thus, in a pure lossy source coding setup, more sophisticated algorithms, for example the one in [17] (in this context for encoding), are required.

Appendix B
Asymptotic Analysis of the Maxwell Decoder

In this appendix, we perform an asymptotic analysis (using DE) of the Maxwell decoder introduced in Section V which ultimately results in the proof of Theorem 3. The derivations parallel and extend those performed in [20] for the Maxwell decoder of LDPC codes on the BEC, highlighting the main differences in the derivations. We assume in the following that the reader is already to some extent familiar with the proof given in [20].

A. Density Evolution Analysis

Let $x^{(\ell)}_0$, $x^{(\ell)}_*$, and $x^{(\ell)}_{\mathrm{g}}$ denote the probability that the second component of a message from a flow node to a counter node at iteration $\ell \ge 0$ is 0, $*$, or g, respectively, according to the update rules of the Maxwell decoder outlined in (20). Likewise, let $y^{(\ell)}_0$, $y^{(\ell)}_*$, and $y^{(\ell)}_{\mathrm{g}}$, $\ell \ge 1$, denote the corresponding probabilities for the second component of a message from a counter node to a flow node, according to the update rules of the Maxwell decoder outlined in (19). The counter node DE updates are

$$\begin{aligned} y^{(\ell)}_0 &= \rho\left(x^{(\ell-1)}_0\right), \\ y^{(\ell)}_* &= 1 - \rho\left(x^{(\ell-1)}_0 + x^{(\ell-1)}_{\mathrm{g}}\right) = 1 - \rho\left(1 - x^{(\ell-1)}_*\right), \\ y^{(\ell)}_{\mathrm{g}} &= 1 - y^{(\ell)}_0 - y^{(\ell)}_*. \end{aligned} \tag{35}$$

On the other hand, the flow node DE updates for odd iterations are

$$\begin{aligned} x^{(2\ell-1)}_0 &= 1 - \lambda\left(y^{(2\ell-1)}_{\mathrm{g}} + y^{(2\ell-1)}_*\right) = 1 - \lambda\left(1 - y^{(2\ell-1)}_0\right), \\ x^{(2\ell-1)}_* &= \lambda\left(y^{(2\ell-1)}_*\right), \\ x^{(2\ell-1)}_{\mathrm{g}} &= 1 - x^{(2\ell-1)}_0 - x^{(2\ell-1)}_*, \end{aligned}$$

and for even iterations

$$\begin{aligned}
x_0^{(2\ell)} &= 1 - \epsilon\lambda\left(y_{\rm g}^{(2\ell)} + y_*^{(2\ell)}\right) = 1 - \epsilon\lambda\left(1 - y_0^{(2\ell)}\right),\\
x_*^{(2\ell)} &= (1-\delta)\epsilon\lambda\left(y_*^{(2\ell)}\right),\\
x_{\rm g}^{(2\ell)} &= 1 - x_0^{(2\ell)} - x_*^{(2\ell)},
\end{aligned}$$

where δ represents the fraction of guesses performed so far. Combining odd and even iterations, we get

$$\begin{aligned}
x_0^{(2\ell)} &= 1 - \epsilon\lambda\left(1 - \rho\left(1 - \lambda\left(1 - \rho\left(x_0^{(2\ell-2)}\right)\right)\right)\right)\\
&= 1 - \epsilon\lambda\left(g(1 - x_0^{(2\ell-2)})\right), \qquad (36)\\
x_*^{(2\ell)} &= (1-\delta)\epsilon\lambda\left(1 - \rho\left(1 - \lambda\left(1 - \rho\left(1 - x_*^{(2\ell-2)}\right)\right)\right)\right)\\
&= (1-\delta)\epsilon\lambda\left(g\left(x_*^{(2\ell-2)}\right)\right), \qquad (37)\\
x_{\rm g}^{(2\ell)} &= 1 - x_0^{(2\ell)} - x_*^{(2\ell)}. \qquad (38)
\end{aligned}$$

To settle the notation, when ℓ tends to infinity, we define

$$\begin{aligned}
\boldsymbol{x} &\triangleq (x_0, x_*, x_{\rm g}) \triangleq \left(x_0^{(\infty)}, x_*^{(\infty)}, x_{\rm g}^{(\infty)}\right),\\
\boldsymbol{y} &\triangleq (y_0, y_*, y_{\rm g}) \triangleq \left(y_0^{(\infty)}, y_*^{(\infty)}, y_{\rm g}^{(\infty)}\right).
\end{aligned}$$

Now, since $(y_0, y_*, y_{\rm g})$ is a function of $(x_0, x_*, x_{\rm g})$ (see (35)), any function of $(\boldsymbol{x}, \boldsymbol{y})$ is indeed just a function of $\boldsymbol{x}$.

Note that the DE recursion in (36)–(38) is for the Maxwell decoder on the equivalent graph introduced Section III. However, according to Proposition 2, the original and equivalent graphs are iteration-by-iteration equivalent for the Maxwell decoder. Thus, we can equivalently base our analysis on the DE recursion in (36)–(38).

B. Analysis

For the Maxwell decoder we choose the following guessing strategy. Set $\delta = 0$ and run the ordinary MP decoding algorithm for CBs (with the update rules in (6) and (7)), until there is no further progress from one iteration round to the next. Then, for each unknown flow node, assign g-messages as outgoing messages to all its adjacent edges with probability $\Delta\delta/(1-\delta)$, after which δ is incremented by $\Delta\delta$. Now, run the Maxwell decoder (with the update rules in (19) and (20)), until it stops. This whole procedure is repeated until $\delta = 1$, or until there are no more unknown flow nodes.

Let $\mathbb{G}$ be the total number of performed guesses and $m_0\Delta\mathbb{G}$ the number of newly guessed flow nodes when δ is incremented by $\Delta\delta$. In more detail, for each $i \in \{1, \dots, m_0\}$, i is chosen independently with probability $\Delta\delta/(1-\delta)$. Then, if the i-th selected flow node is unknown, then assign g-messages as outgoing messages to all its edges and increase the number of newly guessed flow nodes by one. It can be shown that [20]

$$\mathbb{E}[\Delta\mathbb{G}] = \epsilon L(y_*)\Delta\delta. \qquad (39)$$

As explained above, after δ has been incremented by $\Delta\delta$ and g-messages have been assigned as outgoing messages to all edges connected to the $m_0\Delta\mathbb{G}$ newly guessed flow nodes, the Maxwell decoder is run again until no further progress can be made.

For each iteration of the Maxwell decoder there could be flow nodes for which more than one of the incoming messages is a g-message. As explained in Section V, a g-message is just a nonempty list of flow nodes, a corresponding list of coeffcients, and an integer K, giving an explicit resolution rule for the value of the flow node (the value of the flow node is equal to K minus the sum of the values of the flow nodes in the list weighted by their respective coefficients). The fact that there are several incoming g-messages imposes some linear conditions and the number of independent conditions (over all flow nodes) can be lower-bounded using [20, Lem. 11]. As an example, if there are two incoming g-messages to a flow node f_5 and they are represented by the lists $\{\mathsf{f}_1, \mathsf{f}_3, 1, 1, 3\}$ and $\{\mathsf{f}_1, \mathsf{f}_2, \mathsf{f}_4, 1, 1, 1, 2\}$, respectively, then we get the linear condition $3 - \phi(\mathsf{f}_1) - \phi(\mathsf{f}_3) = 2 - \phi(\mathsf{f}_1) - \phi(\mathsf{f}_2) - \phi(\mathsf{f}_4)$, which simplifies to $\phi(\mathsf{f}_3) = 1 + \phi(\mathsf{f}_2) + \phi(\mathsf{f}_4)$, and $\phi(\mathsf{f}_5) = 3 - \phi(\mathsf{f}_1) - \phi(\mathsf{f}_3)$. Let $\mathrm{l}_i^{\rm g}$ be the number of incoming g-messages at a flow node of index i (i.e., f_i), including the flow node itself if it has been guessed. Now, it is not so hard to see that the number of imposed conditions (over all flow nodes) is $\sum_{i=1}^{m_0}(\mathrm{l}_i^{\rm g} - 1)$. On the other hand, some of these conditions may also be dependent. Suppose that there are three incoming g-messages to a counter node (from the flow nodes f_1, f_2, and f_3, respectively) and that they are represented by the lists $\{\mathsf{f}_1, -1, 0\}$, $\{\mathsf{f}_2, -1, 0\}$, and $\{\mathsf{f}_3, -1, 0\}$, respectively, i.e., the i-th g-message indicates that $\phi(\mathsf{f}_i) = 0 - (-1 \cdot \phi(\mathsf{f}_i)) = \phi(\mathsf{f}_i)$, $i = 1, 2, 3$. Then, the g-message from counter node c to flow node f_1 is $\{\mathsf{f}_2, \mathsf{f}_3, 1, 1, \phi(\mathsf{c})\}$, to flow node f_2 is $\{\mathsf{f}_1, \mathsf{f}_3, 1, 1, \phi(\mathsf{c})\}$, and to flow node f_3 is $\{\mathsf{f}_1, \mathsf{f}_2, 1, 1, \phi(\mathsf{c})\}$, from which we get the condition $\phi(\mathsf{c}) = \phi(\mathsf{f}_1) + \phi(\mathsf{f}_2) + \phi(\mathsf{f}_3)$ at all three flow nodes. In particular, using the operational update rule described in the paragraph following (19), the list of flow nodes of the g-message from c to f_1 is the *union* of $\{\mathsf{f}_2\}$ and $\{\mathsf{f}_3\}$, the corresponding list of coefficients is the sum of the corresponding coefficients of the incoming lists multiplied by -1, i.e., $\{-(-1), -(-1)\} = \{1, 1\}$, while the integer K of the outgoing list is $\phi(\mathsf{c})$ minus the sum of all K's of the incoming lists, i.e., $\mathsf{K} = \phi(\mathsf{c}) - (0 + 0) = \phi(\mathsf{c})$, which results in the g-message $\{\mathsf{f}_2, \mathsf{f}_3, 1, 1, \phi(\mathsf{c})\}$ from c to f_1. The other two g-messages from c to f_2 and from c to f_3, respectively, follow by symmetry. Thus, the number of independent imposed linear conditions (over all flow nodes) is in general lower than the number of conditions from the formula $\sum_{i=1}^{m_0}(\mathrm{l}_i^{\rm g} - 1)$. The correction term can be upper-bounded (see [20, Lem. 11]) by $\sum_{\mathsf{c}\in\mathcal{C}_g}(\Gamma(\mathsf{c}) - 1)$, where $\mathcal{C}_g$ is the subset of all counter nodes all of whose incoming messages are g-messages. Note that this is a worst-case situation in which all incoming messages to a counter node are g-messages (see the example above).

Now, we can compute the expected value of the lower bound from [20, Lem. 11] on the number of independent imposed conditions after any stage of the Maxwell decoder (right-hand side of [20, Lemma 11, Eq. (21)], excluding the term $\mathbb{G}$). We perform the same steps as in the analysis in [20, Sec. VI-G]. In fact, the analysis of the Maxwell decoder carried out in [20, Sec. VI-G] is valid also in the context of CBs, except that [20, Eqs. (22) and (23)] need to be properly modified following the DE equations in (36) to (38). In particular, the right-hand side of [20, Eq. (22)] becomes

$$\begin{aligned}
&\epsilon\cdot(1-\delta)\left\{L'(y_*+y_{\mathrm{g}})y_g - L(y_*+y_{\mathrm{g}}) + L(y_*)\right\}\\
&+\epsilon\delta L'(y_*+y_{\mathrm{g}})y_{\mathrm{g}}\\
&= k\left[x_*\left(1-y_*\right)-\left(1-x_0\right)y_0\right] \qquad (40)\\
&\quad -\epsilon\cdot(1-\delta)\left[L(1-y_0)-L(y_*)\right]\\
&\quad + k(1-y_*)x_{\mathrm{g}},
\end{aligned}$$

and the right-hand side of [20, Eq. (23)] becomes

$$\begin{aligned}
&\frac{L'(1)}{\bar{R}'(1)}\left\{\bar{R}'(1-x_*)x_{\mathrm{g}} - \bar{R}(1-x_*) + \bar{R}(1-x_*-x_{\mathrm{g}})\right\}\\
&= \frac{k}{\bar{R}'(1)}\left[-\bar{R}(1-x_*)+\bar{R}(x_0)\right] + k(1-y_*)x_{\mathrm{g}},
\end{aligned} \tag{41}$$

where

$$\bar{R}(x) = \frac{\int_0^x \left(1-g(1-z)\right)\,\mathrm{d}z}{\int_0^1 \left(1-g(1-z)\right)\,\mathrm{d}z}. \tag{42}$$

As in [20], let the function $F(\boldsymbol{x},\epsilon,\delta)$ denote the difference of (40) and (41). We get

$$\begin{aligned}
F(\boldsymbol{x},\epsilon,\delta) &\triangleq k\left[x_*\left(1-y_*\right)-\left(1-x_0\right)y_0\right]\\
&\quad -\epsilon\cdot(1-\delta)\left[L(1-y_0)-L(y_*)\right]\\
&\quad + \frac{k}{\bar{R}'(1)}\left[\bar{R}(1-x_*)-\bar{R}(x_0)\right]\\
&= kx_*\left(1-y_*\right) + \epsilon\cdot(1-\delta)L(y_*) + k\frac{\bar{R}(1-x_*)}{\bar{R}'(1)}\\
&\quad + \bar{F}(x_0,\epsilon,\delta),
\end{aligned}$$

where

$$\bar{F}(x_0,\epsilon,\delta) \triangleq -k(1-x_0)y_0 - \epsilon\cdot(1-\delta)L(1-y_0) - k\frac{\bar{R}(x_0)}{\bar{R}'(1)}$$

depends only on x_0. The function $F(\boldsymbol{x},\epsilon,\delta)$ is the expected value of the number of independent imposed conditions after any stage of the Maxwell decoder. Since $y_* = g(x_*)$ and $\epsilon\cdot(1-\delta)L(g(x_*)) = x_*g(x_*)$ (from (37)), we have

$$\begin{aligned}
F(\boldsymbol{x},\epsilon,\delta) &= kx_*\left(1-g(x_*)\right) + \epsilon\cdot(1-\delta)L(g(x_*))\\
&\quad + k\frac{\bar{R}(1-x_*)}{\bar{R}'(1)} + \bar{F}(x_0,\epsilon,\delta)\\
&= kx_*\left(1-g(x_*)\right) + x_*g(x_*) + k\frac{\bar{R}(1-x_*)}{\bar{R}'(1)}\\
&\quad + \bar{F}(x_0,\epsilon,\delta)\\
&= kx_*\left(1-g(x_*)\right) + x_*g(x_*)\\
&\quad + k\int_0^{1-x_*}\left(1-g\left(1-z\right)\right)\,\mathrm{d}z + \bar{F}(x_0,\epsilon,\delta)\\
&= kx_*\left(1-g(x_*)\right) + x_*g(x_*)\\
&\quad + k\left[\int_0^1\left(1-g\left(z\right)\right)\,\mathrm{d}z - \int_0^{x_*}\left(1-g\left(z\right)\right)\,\mathrm{d}z\right]\\
&\quad + \bar{F}(x_0,\epsilon,\delta)\\
&\overset{(a)}{=} P(x_*) + k\int_0^1\left(1-g\left(z\right)\right)\,\mathrm{d}z + \bar{F}(x_0,\epsilon,\delta),
\end{aligned}$$

where (a) follows from the definition of the trial entropy in Definition 5. Thus,

$$F(\boldsymbol{x}^1,\epsilon,\delta) - F(\boldsymbol{x}^2,\epsilon,\delta) = P(x_*^1) - P(x_*^2)$$

when $\boldsymbol{x}^1 = (x_0^1, x_*^1, x_{\mathrm{g}}^1)$ and $\boldsymbol{x}^2 = (x_0^2, x_*^2, x_{\mathrm{g}}^2)$ have the same x_0-component, i.e., $x_0^1 = x_0^2$. This will become important in the next subsection in the proof of Theorem 3.

Now, if δ is incremented by $\Delta\delta$, the normalized (with respect to the number of flow nodes m_0) expected number of new independent conditions on the values of the newly guessed flow nodes, denoted by $\Delta\mathbb{C}$, can be upper-bounded by the difference

$$\mathbb{E}[\Delta\mathbb{C}] \triangleq F(\boldsymbol{x}(\epsilon,\delta+\Delta\delta),\epsilon,\delta+\Delta\delta) - F(\boldsymbol{x}(\epsilon,\delta),\epsilon,\delta+\Delta\delta). \tag{43}$$

Following [20, Sec. VI-G], we can consider two separate cases:

- $\boldsymbol{x}(\epsilon,\delta)$ is continuous (in the second component) in the interval $[\delta,\delta+\Delta\delta]$, from which it follows (using Taylor series expansion) that

  $$\mathbb{E}[\Delta\mathbb{C}] = O((\Delta\delta)^2), \tag{44}$$

  since it can be shown that the gradient is zero for $\boldsymbol{x} = \boldsymbol{x}(\epsilon,\delta+\Delta\delta)$.
- There is a discontinuity point at $\delta = \delta_j$ where $\delta_j \in [\delta,\delta+\Delta\delta]$, in which case we get

  $$\begin{aligned}
  &\mathbb{E}[\Delta\mathbb{C}] - O(\Delta\delta) = \Delta F_j\\
  &\triangleq F\left(\lim_{\delta\downarrow\delta_j}\boldsymbol{x}(\epsilon,\delta),\epsilon,\delta_j\right) - F\left(\lim_{\delta\uparrow\delta_j}\boldsymbol{x}(\epsilon,\delta),\epsilon,\delta_j\right) \qquad (45)\\
  &\overset{(a)}{=} P\left(\lim_{\delta\downarrow\delta_j}x_*(\epsilon,\delta)\right) - P\left(\lim_{\delta\uparrow\delta_j}x_*(\epsilon,\delta)\right) \qquad (46)\\
  &\overset{(b)}{=} \int_{\lim_{\delta\uparrow\delta_j}x_*(\epsilon,\delta)}^{\lim_{\delta\downarrow\delta_j}x_*(\epsilon,\delta)} h^{\mathrm{EMP}}(z)\,\mathrm{d}\epsilon(z), \qquad (47)
  \end{aligned}$$

  where (a) follows since the x_0-component does not depend on δ (see (36)), and (b) follows from (18).

C. Proof of Theorem 3

We follow the guessing strategy explained above in which for each stage the Maxwell decoder is stuck, δ in incremented by $\Delta\delta$ and then new unknown flow nodes are guessed with probability $\Delta\delta/(1-\delta)$. Assume that at a given stage the Maxwell decoder has reached a fixed-point. The normalized (with respect to the number of flow nodes m_0) number of new guessed flow nodes at this stage is $\Delta\mathbb{G}_\delta$ (see (39)) and the normalized number of new imposed conditions is upper-bounded by $\Delta\mathbb{C}_\delta$ (see (43)). Note that we have made an explicit reference to the specific value of δ by including it as a subscript. We assume the procedure continues until $\delta = 1$ from $\delta = 0$ in small increments of $\Delta\delta$. When the algorithm stops, each assignment of integer values to the guessed flow nodes compatible with the imposed conditions at the end yields a valid flow node size configuration. Thus, we can lower bound the Maxwell EXIT curve as (see [20, Lem. 11])

$$h^{\mathrm{Maxwell}}(\epsilon) \geq \sum_{\delta} \mathbb{E}[\Delta \mathbb{G}_{\delta}] - \sum_{\delta} \mathbb{E}[\Delta \mathbb{C}_{\delta}] \overset{(a)}{=} \int_0^1 \epsilon L(y_*(\epsilon,\delta)) \, \mathrm{d}\delta - \sum_{\delta_j} \Delta F_j + O(\Delta\delta) \tag{48}$$

$$\overset{(b)}{=} \int_0^{\epsilon} L(y(\epsilon')) \, \mathrm{d}\epsilon' - \sum_{\delta_j} \Delta F_j + O(\Delta\delta), \tag{49}$$

where the summations in (48) and (49) are over all discontinuity points δ_j (see (45)). The (a) follows from (39), (44), and (45), while (b) follows from the fact that $y_*(\epsilon,\delta) = y(\epsilon \cdot (1-\delta))$, where $y(\epsilon')$ corresponds to a fixed-point of the standard MP decoder of a CB with a probability of observing a flow of size strictly larger than $f_{\min}$ equal to ϵ' (see (37)).

Now, since $h^{\mathrm{Maxwell}}(\epsilon)$ does not depend of $\Delta\delta$, we can take the limit $\Delta\delta \to 0$ in (49) and we get the lower bound

$$\begin{aligned} h^{\mathrm{Maxwell}}(\epsilon) &\geq \int_0^{\epsilon} L(y(\epsilon')) \, \mathrm{d}\epsilon' - \sum_{\delta_j} \Delta F_j \\ &\overset{(a)}{=} \int_0^{\epsilon} L(y(\epsilon')) \, \mathrm{d}\epsilon' \\ &\quad - \sum_{\delta_j} \int_{\lim_{\delta \uparrow \delta_j} x_*(\epsilon,\delta)}^{\lim_{\delta \downarrow \delta_j} x_*(\epsilon,\delta)} h^{\mathrm{EMP}}(z) \, \mathrm{d}\epsilon(z) \\ &= \int_0^{\epsilon} L(y(\epsilon')) \, \mathrm{d}\epsilon' - \int_{x^{\mathrm{MP}}}^{0} h^{\mathrm{EMP}}(z) \, \mathrm{d}\epsilon(z), \end{aligned} \tag{50}$$

where x^{MP} corresponds to the MP decoding threshold, i.e., $\epsilon(x^{\mathrm{MP}}) = \epsilon_{\mathrm{MP}}$, and (a) follows from (45)–(47). Now, the first term in (50) is just the area under the MP EXIT curve from 0 to ϵ, while the second term is the area under the EMP EXIT curve from the MP decoding threshold to infinity. Since

$$\begin{aligned} \int_0^1 h^{\mathrm{EMP}}(z) \, \mathrm{d}\epsilon(z) &= \int_0^{\epsilon(x^*)} L(y(\epsilon')) \, \mathrm{d}\epsilon' \\ &+ \int_{\epsilon(x^*)}^{\epsilon(1)} L(y(\epsilon')) \, \mathrm{d}\epsilon' - \int_{x^{\mathrm{MP}}}^{0} h^{\mathrm{EMP}}(z) \, \mathrm{d}\epsilon(z), \end{aligned}$$

where x^* is taken from Definition 2, it follows that the right-hand side of (50) (with $\epsilon = \epsilon(x^*)$) simplifies to

$$\begin{aligned} \int_0^{\epsilon(x^*)} & L(y(\epsilon')) \, \mathrm{d}\epsilon' + \int_0^{x^{\mathrm{MP}}} h^{\mathrm{EMP}}(z) \, \mathrm{d}\epsilon(z) \\ &= \int_0^1 h^{\mathrm{EMP}}(z) \, \mathrm{d}\epsilon(z) - \int_{\epsilon(x^*)}^{\epsilon(1)} L(y(\epsilon')) \, \mathrm{d}\epsilon' \\ &\overset{(a)}{=} \int_0^1 h^{\mathrm{EMP}}(z) \, \mathrm{d}\epsilon(z) - \int_{x^*}^1 h^{\mathrm{EMP}}(z) \, \mathrm{d}\epsilon(z) \\ &\overset{(b)}{=} 0. \end{aligned}$$

Equality (a) follows from

$$\int_{\epsilon(x^*)}^{\epsilon(1)} L(y(\epsilon')) \, \mathrm{d}\epsilon' = \int_{x^*}^1 h^{\mathrm{EMP}}(z) \, \mathrm{d}\epsilon(z),$$

which again is true because there is no x' in the interval $(x^*, 1]$ such that $\epsilon(x') = \epsilon(x)$ (see Definition 2), and (b) follows from Definition 2. Now, the statement of the theorem follows, since $h^{\mathrm{Maxwell}}(\epsilon) \geq 0$ for $\epsilon = \epsilon(x^*)$, which again gives the desired upper bound on the Maxwell decoding threshold since there is no x' in the interval $(x^*, 1]$ such that $\epsilon(x') = \epsilon(x^*)$ (which follows from Definition 2).

Appendix C
Proof of Theorem 4

Run the peeling decoder on the original CB graph until it stops. Then, $g(z)$ for the equivalent residual graph, denoted by $\tilde{g}(z;x)$, is given by $\tilde{g}(z;x) = 1 - \tilde{\rho}(z;x)$ where

$$\tilde{\rho}(z;x) = \frac{\tilde{R}'(z;x)}{\tilde{R}'(1;x)},$$

and where

$$\tilde{R}(z;x) = \frac{\bar{R}(1-x+zx) - \bar{R}(1-x) - zx\bar{R}'(1-x)}{1 - \bar{R}(1-x) - x\bar{R}'(1-x)}, \tag{51}$$

x is the largest fixed-point of the DE recursion in (14), and $\bar{R}(z)$ is defined in (42). This can be seen from the analysis of the peeling decoder found in [37, Sec. 2.10]. In particular, the expression for the residual right degree distribution from [26, Eq. (3.124)] is properly adapted to (51).

Now, using (51), after some calculations, we end up with the following simplifed expression

$$\tilde{\rho}(z;x) = 1 - \frac{g(x-zx)}{g(x)}, \tag{52}$$

which immediately leads to the expression given in the theorem for the EMP EXIT curve (defined for the equivalent graph as in (15)) for the expected residual CB graph when the peeling decoder stops.

For the second part of the proof, from Theorem 1, the area under the EMP EXIT curve (for the equivalent graph using the expected counter node degree distribution in (51) with $x = x^*$ (corresponding to the area threshold)) can be written as

$$1 - \tilde{\rho}(0;x^*) + k\left(\tilde{\rho}(0;x^*) - \int_0^1 \tilde{\rho}(1-z;x^*) \, \mathrm{d}z\right), \tag{53}$$

where $\tilde{\rho}(z;x^*)$ is given in (52) with $x = x^*$. Now,

$$\begin{aligned} \int_0^1 \tilde{\rho}(1-z;x^*) \, \mathrm{d}z &= \int_0^1 \tilde{\rho}(z;x^*) \, \mathrm{d}z \\ &= \int_0^1 \left(1 - \frac{g(x^*-zx^*)}{g(x^*)}\right) \mathrm{d}z \\ &= 1 - \frac{1}{g(x^*)} \int_0^1 g(x^*-zx^*) \, \mathrm{d}z \\ &= 1 - \frac{1}{x^* g(x^*)} \int_0^{x^*} g(z) \, \mathrm{d}z. \end{aligned}$$

From the proof of Theorem 2, the fixed-point potential $Q(x)$ from Definition 4 for $x = x^*$ is equal to zero. Thus,

$$\int_0^{x^*} g(z)\ \mathrm{d}z = \left(1 - \frac{1}{k}\right) x^* g(x^*),$$

from which it follows that

$$\int_0^1 \tilde{\rho}(1 - z; x^*)\ \mathrm{d}z = 1 - \frac{1}{x^* g(x^*)} \cdot \left(1 - \frac{1}{k}\right) x^* g(x^*) = \frac{1}{k}.$$

Since $\tilde{\rho}(0; x^*) = 0$, (53) reduces to

$$1 - 0 + k\left(0 - \frac{1}{k}\right) = 0$$

and the result follows.

Acknowledgment

The authors are grateful to Henry D. Pfister for useful discussions and insightful comments.

References

[1] Y. Lu, A. Montanari, B. Prabhakar, S. Dharmapurikar, and A. Kabbani, "Counter braids: A novel counter architecture for per-flow measurement," in *Proc. Int. Conf. Measurement and Modeling of Computer Systems (SIGMETRICS)*, Annapolis, MD, Jun. 2008, pp. 121–132.

[2] Y. Lu, A. Montanari, and B. Prabhakar, "Counter braids: Asymptotic optimality of the message passing decoding algorithm," in *Proc. 46th Annual Allerton Conf. Commun., Control, and Computing*, Monticello, IL, Sep. 2008, pp. 209–216.

[3] ——, "Detailed network measurements using sparse graph counters: The theory," in *Proc. 45th Annual Allerton Conf. Commun., Control, and Computing*, Monticello, IL, Sep. 2007, pp. 1252–1259.

[4] D. L. Donoho and J. Tanner, "Precise undersampling theorems," *Proceedings of the IEEE*, vol. 98, no. 6, pp. 913–924, Jun. 2010.

[5] A. J. Felström 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.

[6] M. Lentmaier, A. Sridharan, D. J. Costello, Jr., and K. S. Zigangirov, "Iterative decoding threshold analysis for LDPC convolutional codes," *IEEE Trans. Inf. Theory*, vol. 56, no. 10, pp. 5274–5289, Oct. 2010.

[7] 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.

[8] S. Moloudi, M. Lentmaier, and A. Graell i Amat, "Spatially coupled turbo codes," in *Proc. Int. Symp. Turbo Codes & Iterative Inf. Processing*, Bremen, Germany, Aug. 2014, pp. 82–86.

[9] ——, "Spatially coupled turbo-like codes," *IEEE Trans. Inf. Theory*, vol. 63, no. 10, pp. 6199–6215, Oct. 2017.

[10] Y. Y. Jian, H. D. Pfister, and K. R. Narayanan, "Approaching capacity at high rates with iterative hard-decision decoding," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Cambridge, MA, Jul. 2012, pp. 2696–2700.

[11] C. Häger, H. D. Pfister, A. Graell i Amat, and F. Brännström, "Deterministic and ensemble-based spatially-coupled product codes," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Barcelona, Spain, Jul. 2016, pp. 2114–2118.

[12] S. Schwandter, A. Graell i Amat, and G. Matz, "Spatially-coupled LDPC codes for decode-and-forward relaying of two correlated sources over the BEC," *IEEE Trans. Commun.*, vol. 62, no. 4, pp. 1324–1337, Apr. 2014.

[13] V. Aref, N. Macris, R. Urbanke, and M. Vuffray, "Lossy source coding via spatially coupled LDGM ensembles," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Cambridge, MA, Jul. 2012, pp. 373 –377.

[14] P. S. Nguyen, A. Yedla, H. D. Pfister, and K. R. Narayanan, "Threshold saturation of spatially-coupled codes on intersymbol-interference channels," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Ottawa, ON, Canada, Jun. 2012, pp. 2181–2186.

[15] K. Takeuchi, T. Tanaka, and T. Kawabata, "Improvement of BP-based CDMA multiuser detection by spatial coupling," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, St. Petersburg, Russia, Jul. / Aug. 2011, pp. 1489–1493.

[16] D. L. Donoho, A. Javanmard, and A. Montanari, "Information-theoretically optimal compressed sensing via spatial coupling and approximate message passing," *IEEE Trans. Inf. Theory*, vol. 59, no. 11, pp. 7434–7464, Nov. 2013.

[17] V. Aref, N. Macris, and M. Vuffray, "Approaching the rate-distortion limit with spatial coupling, belief propagation, and decimation," *IEEE Trans. Inf. Theory*, vol. 61, no. 7, pp. 3954–3979, Jul. 2015.

[18] S. H. Hassani, N. Macris, and R. Urbanke, "Threshold saturation in spatially coupled constraint satisfaction problems," *J. Stat. Phys.*, vol. 150, no. 5, pp. 807–850, Mar. 2013.

[19] A. Yedla, Y.-Y. Jian, P. S. Nguyen, and H. D. Pfister, "A simple proof of Maxwell saturation for coupled scalar recursions," *IEEE Trans. Inf. Theory*, vol. 60, no. 11, pp. 6943–6965, Nov. 2014.

[20] C. Méasson, A. Montanari, and R. Urbanke, "Maxwell construction: The hidden bridge between iterative and maximum a posteriori decoding," *IEEE Trans. Inf. Theory*, vol. 54, no. 12, pp. 5277–5307, Dec. 2008.

[21] V. Chandar, D. Shah, and G. W. Wornell, "A simple message-passing algorithm for compressed sensing," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Austin, TX, Jun. 2010, pp. 1968–1972.

[22] F. R. Kschischang, B. J. Frey, and H.-A. Loeliger, "Factor graphs and the sum-product algorithm," *IEEE Trans. Inf. Theory*, vol. 47, no. 2, pp. 498–519, Feb. 2001.

[23] N. Wiberg, "Codes and decoding on general graphs," Ph.D. dissertation, Dept. of Electrical Engineering, Linköping University, Linköping, Sweden, 1996.

[24] H. Pishro-Nik and F. Fekri, "On decoding of low-density parity-check codes over the binary erasure channel," *IEEE Trans. Inf. Theory*, vol. 50, no. 3, pp. 439–454, Mar. 2004.

[25] E. Rosnes and Ø. Ytrehus, "Turbo decoding on the binary erasure channel: Finite-length analysis and turbo stopping sets," *IEEE Trans. Inf. Theory*, vol. 53, no. 11, pp. 4059–4075, Nov. 2007.

[26] T. Richardson and R. Urbanke, *Modern Coding Theory*. Cambridge, UK: Cambridge University Press, 2007.

[27] Gurobi Inc., "Gurobi," Software Package, 2014, version 6.0.0.

[28] A. Kumar, M. Sung, J. Xu, and J. Wang, "Data streaming algorithms for efficient and accurate estimation of flow size distribution," in *Proc. Joint Int. Conf. Measurement and Modeling of Computer Systems (SIGMETRICS)*, New York, NY, Jun. 2004, pp. 177–188.

[29] E. Rosnes, "Spatially-coupled counter braids," in *Proc. IEEE Inf. Theory Workshop (ITW)*, Hobart, Australia, Nov. 2014, pp. 187–191.

[30] M. R. Sanatkar, Y.-Y. Jian, and H. Pfister, "Practical aspects of spatially-coupled codes," San Diego, CA, Feb. 2014, talk given at *Inf. Theory Appl. (ITA)*.

[31] J. Tanner. Phase transitions of the regular polytopes and cone. [Online]. Available: http://people.maths.ox.ac.uk/tanner/polytopes.shtml

[32] F. Zhang and H. D. Pfister, "Verification decoding of high-rate LDPC codes with applications in compressed sensing," *IEEE Trans. Inf. Theory*, vol. 58, no. 8, pp. 5042–5058, Aug. 2012.

[33] V. Ravanmehr, L. Danjean, B. Vasic, and D. Declercq, "Interval-passing algorithm for non-negative measurement matrices: Performance and reconstruction analysis," *IEEE J. Emerging Sel. Topics Circuits Systems*, vol. 2, no. 3, pp. 424–432, Sep. 2012.

[34] S. Sarvotham, D. Baron, and R. G. Baraniuk, "Compressed sensing reconstruction via belief propagation," Department of Electrical and Computer Engineering, Rice University, Houston, TX, USA, Tech. Rep., Jul. 2006.

[35] H. V. Pham, W. Dai, and O. Milenkovic, "Sublinear compressive sensing reconstruction via belief propagation decoding," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Seoul, Korea, Jun. / Jul. 2009, pp. 674–678.

[36] D. L. Donoho, A. Maleki, and A. Montanari, "Message-passing algorithms for compressed sensing," *Proceedings Nat. Acad. Sci.*, vol. 106, no. 45, pp. 18 914–18 919, Sep. 2009.

[37] C. Méasson, "Conservation laws for coding," Ph.D. dissertation, École Polytechnique Fédérale de Lausanne, Lausanne, Switzerland, 2006.