IRWIN AND JOAN JACOBS
CENTER FOR COMMUNICATION AND INFORMATION TECHNOLOGIES

LDPC Codes for Two-Dimensional Arrays

Yuval Cassuto and Amin Shokrollahi

CCIT Report #849
January 2014

DEPARTMENT OF ELECTRICAL ENGINEERING
TECHNION - ISRAEL INSTITUTE OF TECHNOLOGY, HAIFA 32000, ISRAEL

CCIT REPORT #849 January 2014

LDPC Codes for Two-Dimensional Arrays

Yuval Cassuto, *Member, IEEE,* and Amin Shokrollahi, *Fellow, IEEE*

Abstract

Binary codes over two-dimensional arrays are very useful in data storage, where each array column represents a storage device or unit that may suffer failure. In this paper we propose a new framework for probabilistic construction of codes on two-dimensional arrays. Instead of a pure combinatorial erasure model used in traditional array codes, we propose a mixed combinatorial-probabilistic model of limiting the number of column failures, and assuming a binary erasure channel in each failing column. For this model we give code constructions and detailed analysis that allow sustaining a large number of column failures with graceful degradation in the fraction of erasures correctable in failing columns. Another advantage of the new framework is that it uses low complexity iterative decoding. The key component in the analysis of the new codes is to analyze the decoding graphs induced by the failed columns, and infer the decoding performance as a function of the code design parameters, as well as the array size and failure parameters. A particularly interesting class of codes, called probabilistically MDS array codes, gives fault-tolerance that is equivalent to traditional MDS array codes. The results also include a proof that the two-dimensional codes outperform standard one-dimensional LDPC codes.

I. Introduction

Linear codes constructed from low density matrices are the keystone of contemporary coding theory, with huge impact on both theoretical research and practical applications. The sparsity of the code matrices offers a compelling complexity advantage in implementation, and achieving this low complexity with optimality in redundancy is the great achievement of a large body of deep research. Interestingly, two separate coding-theory fields aim at the above objective of developing low-density codes with good information efficiency.

One is the field of *array codes* [3]. Array codes, defined over two-dimensional binary arrays, use low-density parity-check matrices to construct codes for erasures or errors of full columns. The low-density property offers complexity benefits in encoding and decoding operations. Hence the task of array codes is to combat column-level erasures/errors, while being defined by operations over smaller information units (bits or small groups of bits). The ultimate goal of array-code research is code families that are Maximum Distance Separable (MDS) from the column perspective, thus having optimal redundancy for a given column correction requirement. Array codes are widely used in practice, being employed as a central element of RAID (Redundant Arrays of Inexpensive Disks) [14] storage systems. Each array column then represents an individual storage device, whose failure is modeled as a column erasure.

The second, and better-known low-density coding field, is *low-density parity-check (LDPC) codes under iterative decoding* [15]. Without need for detailed introduction, in this area the objective is to construct low-density (one dimensional) codes that will decode well under sub-optimal iterative decoding algorithms.

So far, despite the structural similarities, the two low-density coding fields evolved in essential separation. Array codes have concentrated on algebraic constructions and decoding algorithms that guarantee fixed numbers of column erasures over small array dimensions. In contrast, the theory of iteratively decoded low-density parity-check codes has sought probabilistic code constructions that with high probability have good iterative decoding performance in the limit of large block lengths. Previous work exists where iterative decoding of known array codes over one-dimensional channels is experimentally examined [9], [2], but no attempt has been made (to the best of our knowledge) to construct new array codes for iterative decoding over two-dimensional channels. Such constructions are highly motivated by the current state of matters in array-code theory and practice. Algebraic array codes rigidly assume that columns are erased at full, while in practice many storage devices have failure modes that render only part of their data inaccessible. This strong error model introduces significant complexity penalties, with encoding and decoding complexities that steeply grow with the number of column erasures. Moreover, algebraic array codes guarantee correction of a certain number of column erasures, with a sharp transition to failure if the specified number of column erasures is exceeded. These issues have triggered a recent line of work constructing array codes for combinations of full-column and individual-symbol erasures (and errors) [4], [16].

For the same purpose of targeting more realistic erasures in storage arrays, the study in this paper proposes the first framework for construction and analysis of two-dimensional array codes under iterative decoding. The new framework merges the fields of array coding and LDPC coding in the sense that it applies the deep construction and analysis tools of LDPC codes to an error model very natural for array codes. The outcome from this framework are codes that can extend the fault tolerance of storage arrays with complexities significantly lower than what algebraic array codes can achieve. The new framework comprises three main components

1) An erasure model that combines an integer bound on the number of failing columns with an erasure probability within failing columns.

Yuval Cassuto is with the Department of Electrical Engineering, Technion – Israel Institute of Technology, Haifa Israel (email: ycassuto@ee.technion.ac.il). Part of the research was done while at the School of Computer and Communication Sciences, École Polytechnique Fédérale de Lausanne (EPFL).

Amin Shokrollahi is with the School of Computer and Communication Sciences, École Polytechnique Fédérale de Lausanne (EPFL), Station 14, CH-1015 Lausanne, Switzerland (e-mail: amin.shokrollahi@epfl.ch). Part of the results of this paper appeared in the proceedings of the 2011 International Symposium on Information Theory [8].

2) A probabilistic code construction that considers the two-dimensional array structure when choosing parity groups.
3) Performance analysis using the decoding graphs induced by erasure events under the model in 1.

The majority of the paper is devoted to the theoretical performance analysis of the new codes over the new channel model. The main challenge to analysis in this framework is the introduction of combinatorial objects (originating from the integer number of failing columns) into the probabilistic analysis of code ensembles under random erasures. Due to the non-uniformity of erasure probability across the array bits, the graph observed by the decoder is substantially different from the graph of the designed code. Hence analysis must accommodate the parameters of array size and failure cardinality in addition to the parameters of the code itself.

We now detail the paper's content and main contributions. Section II defines the coding structure as a two-dimensional array with dimensions $b \times n$. The erasure model for the array allows up to r (out of the n) columns to fail, and in each failing column bits are erased i.i.d. with probability ϵ. The number of columns n is assumed to be a parameter of the code, and for the sake of analysis, the column size b is assumed to grow to infinity. This mirrors practical storage systems having a given number of nodes, but whose individual storage capacities are large (and constantly growing). In Section III we give code constructions for the two-dimensional erasure model. The constructions are given as code ensembles, from which codes are drawn at random. The code ensembles have the same flavor as standard one-dimensional LDPC ensembles, only with an important additional property of allowing a check node to have *at most one neighbor from each column*. This property is the key to the remainder of the results in the paper. In Section IV the codes are analyzed over the two-dimensional erasure model. The analysis is centered at an object we call the decoder *induced graph*. The induced graph is a subgraph of the code graph, obtained by removing the variable nodes in non-failing columns. Unlike one-dimensional codes, to analyze the performance of the code it is not sufficient to consider the designed code graph. Rather, the performance will be determined by the interplay between the code graph (which we can control) and a transformation induced by the combinatorial process of taking r out of the n columns (which is dictated to us by the model). Section IV reveals the most attractive feature of LDPC array codes: their ability to sustain a large number of column failures, where the in-column erasure probability ϵ that they can sustain degrades gracefully with r. This feature is unique to our scheme, in contrast to the rigid r offered by traditional array codes. Section V is devoted to a special class of codes correcting an $\epsilon = 1$ fraction of erasures in r failing columns, with optimal redundancy of r/n. The importance of this class of codes is that they offer fault tolerance that is equivalent to traditional MDS (maximum distance separable) array codes. Then in Section VI we prove analytically that two-dimensional LDPC ensembles are superior to one-dimensional ones for the studied erasure model. The way we establish this result is through explicit rate bounds for one-dimensional and two-dimensional codes, which are proved to exhibit a gap in favor of two-dimensional codes.

II. Two-Dimensional Erasure Model

In the main target application for this work – arrays of storage devices – erasures are not uniformly distributed across the two-dimensional array. Rather, a few of the columns, corresponding to failing devices, will have a large number of erasures, and the rest of the columns, corresponding to non-failing devices, will have no erasures at all. We now seek to define a general two-dimensional erasure model that captures this bi-modality of erasure probability. But before considering the characterization of erasures over the array, we need to define the structure of the array itself.

bn bits are organized in a two dimensional array $A = (a_{i,j})$, $1 \leqslant i \leqslant b$, $1 \leqslant j \leqslant n$. Note that in practice each $a_{i,j}$ may not be a single bit, but a larger information unit. Nevertheless, we assume the basic array unit to be a bit throughout the paper, since XOR operations over bits can be easily extended to larger sets of bits. Since the total capacity of the storage device is much larger than the desirable unit of XOR operations for the code, the column size (number of XOR units per device) b will be assumed large, and growing to infinity for the purpose of analysis. The number of columns n will however be assumed a fixed integer. This array structure is natural for real storage systems, in which the number of devices stay fixed at the order of roughly 10s of devices, while the capacities of constituent devices grow rapidly with the scaling of storage densities. This chosen array structure is depicted in Figure 1.

A. Mixed probabilistic-combinatorial erasure model

Let F_i be an indicator function for the event that column i is in failure state.

$$F_i = \begin{cases} 1 & \text{if column } i \text{ fails} \\ 0 & \text{otherwise} \end{cases}$$

In a failing column, every bit is erased independently and with equal probability. In a non-failing column, none of the bits are erased. Formally, the erasure probability of bits in column i is specified as a function of F_i:

$$\epsilon_i = \begin{cases} \epsilon & \text{if } F_i\text{=1} \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

where ϵ is a global erasure probability applying to all failed columns.

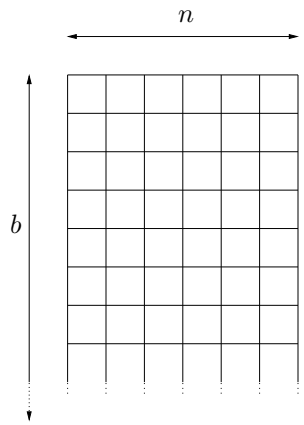

Figure 1. Two-dimensional array with dimensions $b \times n$, where n is fixed and $b \to \infty$.

To the probabilistic intra-column erasure model above we now add a combinatorial column-failure model. Within an array of n columns, at most r columns are failing. Hence

$$|\{i : F_i = 1\}| \leqslant r. \tag{2}$$

The rest of the paper is thrust toward the correction of erasures that fall under the model defined jointly by (1) and (2).

III. Array Code Ensembles

The usual approach to combat erasures in two-dimensional arrays is to algebraically construct array codes for a prescribed number of column failures r (see for example [5], [1]; and [7] for a more general column erasure model). Algebraic array codes are defined explicitly, by specifying sets of array locations with restricted parity values. In addition, algebraic array codes aim at the extreme case of $\epsilon = 1$, i.e. *all* bits are assumed erased in a failing column. In this paper, in contrast, array codes are constructed *probabilistically*, by specifying ensembles of parity restrictions over the array. Moreover, the case of interest here is $\epsilon < 1$, which better describes realistic failure modes in storage systems.

For the two-dimensional erasure model specified by (1) and (2), we now propose a probabilistic array-code construction. The resulting codes are two-dimensional variants of the well known LDPC codes, called herein *LDPC array codes* or (interchangeably) *two-dimensional LDPC codes*.

Definition 1. [LDPC Array Codes/2D LDPC Codes] *An LDPC array code for array $A = (a_{i,j})$ consists of parity constraints of the form*

$$a_{i_1,j_1} + a_{i_2,j_2} + \cdots + a_{i_d,j_d} = 0,$$

where $+$ represents the binary XOR operation, and $d \leqslant n$ is the degree of the parity constraint. The code has the following properties.

1) *The parity-constraint degrees d and the array bits a_{i_m,j_m} in each constraint are randomly selected according to some probability distribution.*
2) *A parity constraint has at most one bit from each array column.*

The key feature of Definition 1 is Property 2 above, assuring that no two or more bits from the same column appear in a parity constraint. The reasoning behind this feature is that multiple bits from the same column cause correlated erasures within a parity constraint in case this column fails. The benefits of this feature are made more precise in Section VI.

In Figure 2 are illustrated two sample parity constraints: $a_{2,1} + a_{3,2} + a_{4,3} + a_{2,4} = 0$ (left XOR node) and $a_{3,1} + a_{1,2} + a_{1,3} + a_{5,4} = 0$ (right XOR node). As required by the definition of LDPC array codes, the j indices (column locations) in both parity constraints are unique. On the basis of the above definition of LDPC array codes, we proceed to define the codes as Tanner graphs sampled from array-code ensembles.

A. Regular array-code ensembles

An array-code ensemble is called (l, d)-regular if every array bit participates in l parity constraints, and every parity constraint consists of d array bits, each from a distinct column. Array codes are constructed from array-code ensembles by randomly sampling $|E| = nbl$ edges in a bipartite graph with $N = nb$ variable nodes and $M = nbl/d$ check nodes (assuming d divides nbl). The number of outgoing edges from the variable nodes and from the check nodes are called variable-node and check-node *degrees*, respectively. The sampling process differs from the standard graph sampling process of one-dimensional LDPC codes in the restriction on check nodes to have at most one variable-node neighbor from any column. The (l, d)-regular array-code ensemble is formally defined in the following construction.

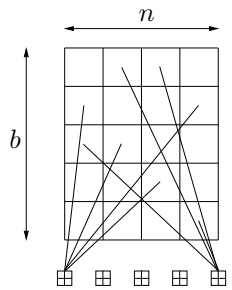

Figure 2. Array parity constraints. Two parity constraints with $d = n = 4$ are shown (each column contributes exactly one bit to each of the sample parity constraints).

Construction 1. [(l, d)-regular array-code ensemble] *Let V_1 be a set of $N = nb$ variable nodes from a $b \times n$ array, each with l sockets. Let V_2 be a set of M check nodes, each with d sockets. The total number of sockets in V_1 nodes and the total number of sockets in V_2 nodes each equals $|E| = Nl = Md$. For each of the V_1 and V_2 nodes we label the sockets with the set $[|E|] = \{1, \ldots, |E|\}$. Define Σ to be the set of permutations on $[|E|]$. Define $\Sigma' \subset \Sigma$ to be the set of permutations on $[|E|]$ that do not connect two or more sockets of a check node to variable nodes in the same column. An (l, d)-regular array-code ensemble consists of all bipartite graphs obtained by connecting the V_1 and V_2 sockets by permutations in Σ', with the uniform probability distribution on the set Σ'.*

B. Irregular array-code ensembles

As in one-dimensional LDPC codes, better iterative-decoding performance can be achieved when the degree regularity constraint is lifted [12]. To specify the ensemble degree distributions, we adopt the standard notation used for one-dimensional LDPC codes [15]. This notation will be extended and refined in the next sections for the analysis of array-code ensembles. For each $i \in \{1, \ldots, \mathtt{l}_{\max}\}$, we denote by L_i the probability that a variable node has degree i. Similarly, for each $i \in \{1, \ldots, \mathtt{d}_{\max}\}$, we denote by R_i the probability that a check node has degree i. When the L_i and R_i probabilities are viewed as coefficients of polynomials, we get the variable-degree distribution polynomial $L(x) = \sum_{i=1}^{\mathtt{l}_{\max}} L_i x^i$, and the check-degree distribution polynomial $R(x) = \sum_{i=1}^{\mathtt{d}_{\max}} R_i x^i$. Since iterative decoding is analyzed through messages passed on edges, similar distributions from edge perspective are often useful. These are $\lambda(x) = \sum_{i=1}^{\mathtt{l}_{\max}} \lambda_i x^{i-1}$ and $\rho(x) = \sum_{i=1}^{\mathtt{d}_{\max}} \rho_i x^{i-1}$. λ_i is the probability that an edge is connected to a variable node with degree i. ρ_i is the probability that an edge is connected to a check node with degree i. The relations between the node-perspective and edge-perspective degree distributions are given by $\lambda(x) = L'(x)/L'(1)$ and $\rho(x) = R'(x)/R'(1)$, where the operator $'$ represent the function's first derivative.
As in the one-dimensional case, a simple generalization of the regular construction in Construction 1 gives a construction for irregular codes.

Construction 2. [$L(x), R(x)$ irregular array-code ensemble] *Let V_1 be a set of $N = nb$ variable nodes from a $b \times n$ array. For each i, NL_i (assumed to be an integer) variable nodes from V_1 have i sockets each. Let V_2 be a set of M check nodes. For each i, MR_i (assumed to be an integer) check nodes from V_2 have i sockets each. The total number of sockets in V_1 nodes and the total number of sockets in V_2 nodes each equals $|E| = NL'(1) = MR'(1)$. The $|E|$ sockets on the V_1 side are connected to the $|E|$ sockets on the V_2 side by a uniformly selected permutation from Σ' (defined in Construction 1 as the set of permutations that do not connect two or more sockets of a check node to variable nodes in the same column).*

The regular construction of Construction 1 is clearly a special case of Construction 2 with $L(x) = x^l$ and $R(x) = x^d$.

IV. Analysis of Iterative Erasure Decoding

A natural method to decode array codes from the proposed ensembles is by iterative message-passing decoding, also called belief propagation (BP) decoding. This is an especially simple standard way to decode one-dimensional LDPC codes over erasure channels. The special property of LDPC array codes is that the variable nodes in the code graph are partitioned into two sets: variables in failed columns with erasure probability ϵ, and variables in non-failed columns with erasure probability 0. This property has little impact on the decoder implementation, but a significant impact on the decoding analysis. The decoding-analysis framework developed here for array-code ensembles follows the adaptation of established tools from one-dimensional LDPC ensembles [15] to the special structure of array-code decoding graphs. In particular, our main objective is to obtain BP *decoding thresholds* for LDPC array codes under the two-dimensional erasure model. For a given array-code ensemble, there is no longer a single BP threshold, but one for each number of failing columns. Given r failed columns, ϵ_r is defined such that decoding succeeds with high probability as long as the failed-column erasure probability satisfies $\epsilon < \epsilon_r$. In the sequel when r is clear from the context, we often omit the index r from ϵ_r.

A. Induced decoding graphs

Throughout the analysis, we will assume that exactly r columns are failing. Figure 3 reveals the relations between the code graph and the decoding graph induced by the r failed columns. As usual, variable nodes are represented as circles, and check nodes as squares. A variable node belonging to a failed column is marked black, and its erasure probability is ϵ. A variable node in a non-failing column is marked white, and its erasure probability is 0. White variable nodes are completely known, and thus do not appear in the decoding graph. As a result of removing the white variable nodes, check nodes may have lower degrees in the decoding graph compared to their degrees in the code graph – cf. Figure 3(a). Check nodes all of whose neighbors are white, are also marked white and removed from the decoding graph – cf. Figure 3(b). It is observed that black variable nodes always have the same degree in the code and decoding graphs, because the only removed check nodes are those with all-white variable neighbors – cf. Figure 3(c). The structure of decoding graphs induced by r column failures

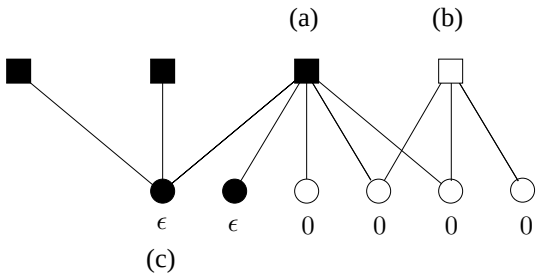

Figure 3. Sample from a code/decoding graph of an LDPC array code. Black variable nodes (circles) are from failed columns; white variable nodes are from non-failed columns. The white variable nodes can be removed from the decoding graph because they have erasure probability 0. (a) a check node with degree 5 in the code graph and degree 2 in the decoding graph. (b) a check node is removed from the decoding graph if all its variable neighbors are white. (c) a black variable node always has the same degree in the decoding graph as in the code graph.

is made more precise with the following definitions.

For a given LDPC array code and a given set of failed columns, we define the decoding graph induced by the failed-column set.

Definition 2. *Given a code graph $\mathcal{G}$ and a set of failing columns $\mathcal{R} = \{i_1, \ldots, i_r\}$, we define the* **induced decoding graph** *$\mathcal{G}^{\mathcal{R}}$ as the subgraph of $\mathcal{G}$ containing the variable nodes that reside in the columns of $\mathcal{R}$.*

The induced decoding graph will play a central role in the analysis of array-code ensembles. In the sequel, we seek to characterize the probabilistic structure of induced decoding graphs. The principal question is to find the distribution of induced decoding graphs given the code-graph distribution and the number of column failures. A nice outcome from this approach is that established analysis tools, such as binary erasure channel (BEC) density evolution, can be applied to two-dimensional codes by considering the induced decoding graph rather than the code graph itself.

Recall [15] that given an ensemble of one-dimensional LDPC code graphs, the BEC density-evolution threshold (*decoding threshold* for short) is the supremum of $\epsilon \in [0,1]$ such that over a BEC(ϵ) channel the failure probability of a BP decoder tends to zero as the block size tends to infinity. We now want to extend the decoding-threshold analysis to array-code ensembles. For this purpose, we first characterize the probabilistic effect of removing variable nodes in non-failing columns from the decoding graph. We start with the following definitions.

Definition 3. *Given design degree distributions of variable resp. check nodes $L(x),\ R(x)$ (from node perspective) or $\lambda(x),\ \rho(x)$ (from edge perspective), the* **induced degree distributions** *are the degree distributions of an induced decoding graph containing only the variable nodes belonging to r array columns. We denote them as*

$$\tilde{L}(x),\ \tilde{R}(x)$$

(from node perspective) and

$$\tilde{\lambda}(x),\ \tilde{\rho}(x)$$

(from edge perspective). We note that given the probabilistic nature of Construction 2, the induced distributions depend only on r, and not on the identity of the r failed columns in $\mathcal{R}$. For notational convenience, we keep this dependence on r implicit, in cases where the value of r is clear from the context.

Because variable nodes maintain their code degree in the induced graph (Figure 3 (c)), we always have $\tilde{L}(x) = L(x)$. Similarly to the code-graph degree distributions, the induced edge-perspective distributions are related to the node-perspective distributions by $\tilde{\lambda}(x) = \tilde{L}'(x)/\tilde{L}'(1)$ and $\tilde{\rho}(x) = \tilde{R}'(x)/\tilde{R}'(1)$.

An obvious generalization of one-dimensional density evolution [11] gives that the decoding threshold ϵ_r is the largest ϵ such that

$$x > \epsilon\tilde{\lambda}(1 - \tilde{\rho}(1-x)),\ \forall x \in (0,1], \tag{3}$$

for the $\tilde{\lambda}(x),\ \tilde{\rho}(x)$ given in Definition 3. We can also define the induced rate of the decoding graph, which is different from (lower than) the actual code rate.

Definition 4. *Given induced variable resp. check degree distributions $\tilde{\lambda}(x),\ \tilde{\rho}(x)$, we define the* **induced rate** *as*

$$1 - \frac{\int_0^1 \tilde{\rho}(x)dx}{\int_0^1 \tilde{\lambda}(x)dx}.$$

Explicit calculation of the induced degree distributions of array-code ensembles will be the main tool for analysis and design of LDPC array codes in the remainder of the paper.

B. Regular codes with degree-n check nodes

An interesting (and simple) special case of LDPC array codes is when the code is regular with check degree $d = n$. The restriction on the graph connectivity in Construction 2 implies that given check-regularity of $d = n$, every check node has exactly one variable-node neighbor from each array column. It is not hard to see that for this case the induced code graph is regular as well, but with a check-degree r instead of n. As a result, the following theorem can be proved.

Theorem 1. *An (l, n)-regular array-code ensemble has design rate $1 - l/n$, and an erasure decoding threshold of an (l, r)-regular one-dimensional LDPC code ensemble, for any set of r failed columns.*

Proof: The argument regarding the design rate is identical to (l, n)-regular one-dimensional LDPC codes. We now prove the statement on the decoding threshold. After removing all $(n - r)b$ variable nodes in non-failing columns, the code graph becomes an (l, r)-regular graph, with the restriction that each of the r neighbors of a check node comes from a different column. For one-dimensional LDPC ensembles, it is well known [15, Ch.3] that after a finite number ℓ of decoding iterations, the computation graph of a BP decoder around a given variable node is a tree with probability tending to 1 as the number of variable nodes tends to infinity, giving rise to the condition (3). It thus remains to prove that this property still holds given the restricted sampling of Construction 1. To prove this, we make the (incorrect) assumption that all edges connect to variable nodes in the same column. While this is not the case for the code construction, this assumption can only increase the probability that the computation graph has cycles, and thus it is valid for proving the acyclic property. In that case, the random permutation with column restrictions maps to uniformly distributed permutations on the sockets of b variable nodes in one column. This is identical to the uniform permutation sampling in the one-dimensional case. Since the column size b tends to infinity, the one-dimensional result carries over to the two-dimensional case. ■

The nice consequence of Theorem 1 is that array codes with high rate (thanks to the high check-node degrees in the *code graph*) have high thresholds (thanks to the low check-node degrees in the *induced decoding graph*). Moreover, the decoding threshold gracefully decreases as the number r of failed columns grows. Example 1 shows the decoding performance of a regular check-node degree n array code ensemble, and compares it with an MDS array code of the same rate.

Example 1. *Suppose we want to design an array code for $r = 3$ column failures. One alternative is to use a traditional MDS array code with rate $1 - 3/n$. Another alternative is to use a $(3, n)$-regular array-code ensemble, which has the same rate. In Figure 4 the thresholds of both options are plotted for $1 \leqslant r \leqslant 7$. For the MDS code, the threshold is either 1 for $r \leqslant 3$, or undefined for $r > 3$ (square markers). On the other hand, the array-code ensemble can tolerate many erasures even for large r values (dot markers). For $r = 3$ there is a gap in favor of the MDS code, where the array-code ensemble only achieves a threshold of 0.844, strictly smaller than the 1 threshold of the MDS array code. The problem with the regular ensemble is that at $r = 3$ we get a threshold of a $(3, 3)$-regular induced graph, which is not an effective rate 0 code. Tightening the gap necessitates the departure from the (l, n)-regular ensemble family, into more general ensembles discussed in the next sub-sections.*

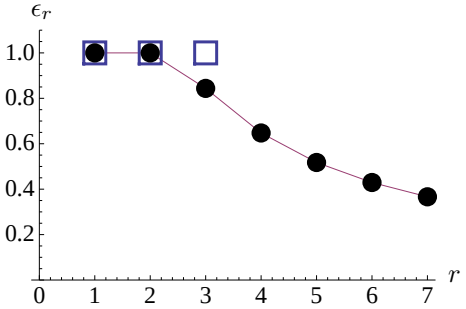

Figure 4. Threshold values as a function of the number of failing columns. dots: $(3, n)$-regular ensembles, squares: traditional MDS array codes.

C. Regular codes with general check-node degrees

In the previous sub-section, a code graph with regular check-degree n induced a regular check-degree r in the decoding graph. Now we want to consider the more general case where check nodes have a general – but still constant – degree $d \leqslant n$. As we will see, the induced graph in this case is no longer regular.

Given a set of r failed columns, a check node has between 0 and r neighbors in the induced graph (assuming $r \leqslant d$). From symmetry in the graph sampling of LDPC array codes (specified in Construction 1), the sets of d distinct columns neighboring a check node are uniformly distributed among the size-d subsets of $\{1, \ldots, n\}$. Hence the induced degree distribution is given by

$$\Pr(\text{degree } i) = \frac{\binom{r}{i}\binom{n-r}{d-i}}{\binom{n}{d}} \triangleq T_i^{(n,r,d)}. \tag{4}$$

The numerator is the number of size-d subsets partitioned to i failed columns and $d-i$ non-failed columns. The denominator is the total number of size-d subsets. Hence given a uniform column sampling of d check-node edges, the induced degrees are distributed according to the *hypergeometric distribution* [10] $T^{(n,r,d)}$. It is clear that when $d = n$, $T_i^{(n,r,n)}$ degenerates to a delta function $\delta[i-r]$, i.e., $\Pr(\text{degree } i)$ equals to 1 when $i = r$ and 0 otherwise (giving the regularity proved in Theorem 1). Note that if $r + d \leqslant n$, there is a non-zero probability that a check node has degree 0 in the induced graph, in which case it is removed from the graph. Recall from Section IV-A that the induced graph remains regular with respect to the variable-node degrees. Altogether we obtain the following result.

Theorem 2. *An (l, d)-regular array-code ensemble has design rate $1 - l/d$ and an erasure decoding threshold of an irregular one-dimensional LDPC code ensemble with variable resp. check degree distributions (from node perspective) given by*

$$\tilde{L}(x) = x^l \quad , \quad \tilde{R}(x) = \sum_{i=0}^{r} T_i^{(n,r,d)} x^i.$$

Proof: From (4), the check-degree distribution of the induced graph is given by $\tilde{R}(x)$. The degree distribution of variable nodes is unchanged in the induced graph, since removed (induced degree 0) check nodes affect only variable nodes outside the induced graph. To prove that finite-depth computation graphs are trees with high probability we essentially repeat the proof of Theorem 1. ■

Note that $\tilde{R}_i \neq 0$ only if $d - (n - r) \leqslant i \leqslant r$. The right inequality trivially follows from the fact that there are no more than r non-failing columns for a check node to be connected to. The left inequality follows from the fact that at most $n - r$ edges out of the check node's d edges are connected to variable nodes in non-failing columns.

Check degree $d < n$ can improve decoding performance over codes with check degree $d = n$ with the same rate. This is because a lower check degree allows to reduce the variable degree, which in general results in better BP thresholds. Example 2 compares two such codes.

Example 2. *For $n = 16$, we compare two possible ensembles: the $(3, 12)$ and $(4, 16 = n)$ regular ensembles. Both have the same $\text{rate} = 3/4$. Figure 5 plots the decoding thresholds for $3 \leqslant r \leqslant 6$. The $(3, 12)$ ensemble (square markers) gives better thresholds for every r compared to the $(4, 16)$ ensemble (dot markers).*

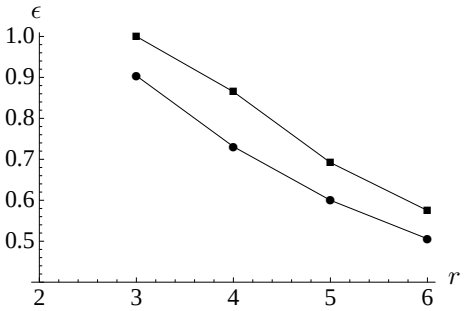

Figure 5. Threshold value comparison between degree n (dots) and degree $d < n$ (squares) ensembles.

Note that unlike check degree n regular codes whose thresholds are only a function of l, codes with check-degree $d < n$ have thresholds that depend on the length n as well, due to the $T_i^{(n,r,d)}$ coefficients in their induced degree distributions, which depend on n.

D. Irregular codes

The most general array-code ensembles have irregular degrees in the original code graph as well, and not only in the induced graph as in the previous sub-section. We now examine the induced decoding graphs of such ensembles. Let $L(x) = \sum_{l=1}^{\mathtt{l}_{\max}} L_l x^l$ and $R(x) = \sum_{d=1}^{\mathtt{d}_{\max}} R_d x^d$ be the node-perspective variable and check degree distributions, respectively. A given check node with degree d will have induced degree i according to the distribution in (4). Now considering all possible degrees d in the degree distribution $R(x)$, we obtain the induced check-degree distribution

$$\tilde{R}(x) = \sum_{d=1}^{\mathtt{d}_{\max}} R_d \sum_{i=0}^{d} T_i^{(n,r,d)} x^i = \sum_{i=0}^{\mathtt{d}_{\max}} x^i \sum_{d=i}^{\mathtt{d}_{\max}} R_d \cdot T_i^{(n,r,d)},$$

where the last equality is obtained by reversing the order of summation. Therefore, the i^{th} coefficient of the induced check-degree distribution equals in the irregular case

$$\tilde{R}_i = \sum_{d=1}^{\mathtt{d}_{\max}} R_d \cdot T_i^{(n,r,d)} = \sum_{d=1}^{\mathtt{d}_{\max}} R_d \frac{\binom{r}{i}\binom{n-r}{d-i}}{\binom{n}{d}}. \tag{5}$$

(Starting the summation at $d = 1$ instead of $d = i$ simplifies the expression without changing the sum.) The expression above leads to the following theorem.

Theorem 3. *An $(L(x), R(x))$-irregular array-code ensemble has design rate $1 - L'(1)/R'(1)$ and an erasure decoding threshold of an irregular one-dimensional LDPC code ensemble with variable and check degree distributions*

$$\tilde{L}(x) = L(x) \quad , \quad \tilde{R}(x) = \sum_{i=0}^{r} \left(\sum_{d=1}^{\mathtt{d}_{\max}} R_d \cdot T_i^{(n,r,d)} \right) x^i.$$

Proof: The induced check-degree distribution $\tilde{R}(x)$ is proved in the preceding analysis. As in the regular case, the variable degree distribution is unchanged in the induced graph, since removed check nodes affect only variable nodes outside the induced graph. Convergence to tree ensembles is proved identically to Theorem 1. ■

The $T_i^{(n,r,d)}$ coefficients thus induce a linear transformation on the design check-degree distribution, as formalized in the sequel. Let $\tilde{R} = [\tilde{R}_0, \ldots, \tilde{R}_r]$ be the vector of induced degree-distribution probabilities. When the inducing distribution $T_i^{(n,r,d)}$ is viewed as a $\mathtt{d}_{\max} \times (r+1)$ matrix $T^{n,r} = \{t_{d,i}\}$, then the induced check-degree distribution is obtained from the design check-degree distribution $R = [R_1, \ldots, R_{\mathtt{d}_{\max}}]$ by the linear transformation

$$\tilde{R} = R \cdot T^{n,r}. \tag{6}$$

V. Probabilistically MDS Array Codes

Array-code ensembles are better than r-erasure-correcting MDS codes in tolerating more than r column failures. But so far in the paper, array-code ensembles failed to match the threshold of $\epsilon_r = 1$ that MDS codes have with $\text{rate} = 1 - r/n$ (cf. Figure 4). It is thus the purpose of this section to close this gap and provide array-code ensembles with $\text{rate} = 1 - r/n$ and a threshold approaching $\epsilon_r = 1$. Such codes are called herein *probabilistically MDS* codes.

Definition 5. *An array-code ensemble is* **probabilistically MDS** *if for some integer r, the ensemble rate equals $1 - r/n$, and for any r failing columns it can recover with high probability from any erasure probability $\epsilon_r < 1$.*

It is clear that tolerating arbitrary erasure probabilities $\epsilon_r < 1$ in r columns cannot be achieved with redundancy smaller than r/n. Hence similarly to standard MDS codes, probabilistically MDS codes attain the optimal redundancy for their erasure-correction capabilities. Formulating the probabilistically MDS property in terms of the threshold gives the following proposition.

Proposition 4. *An array-code ensemble is probabilistically MDS if and only if it has rate $1 - r/n$ and its induced degree distributions satisfy*

$$\tilde{\lambda}(1 - \tilde{\rho}(1 - x)) = x, \ \forall x \in (0, 1]. \tag{7}$$

Proof: If $\tilde{\lambda}(1 - \tilde{\rho}(1 - x)) = x$, then multiplying the left-hand side by any $\epsilon_r < 1$ will give $\epsilon_r \tilde{\lambda}(1 - \tilde{\rho}(1 - x)) < x$, as required by the threshold definition of (3). To show necessity, suppose that $\tilde{\lambda}(1 - \tilde{\rho}(1 - x_0)) > x_0$ for some $x_0 \in (0, 1]$. Then by substituting $\epsilon_r = x_0 / \tilde{\lambda}(1 - \tilde{\rho}(1 - x_0))$ we get that $\epsilon_r \tilde{\lambda}(1 - \tilde{\rho}(1 - x_0)) = x_0$, violating the threshold definition of (3). ■

The existence of explicit probabilistically MDS array-code ensembles is proved in the following sub-sections.

A. Check-degree n probabilistically MDS codes

The first probabilistically MDS code construction we present is a check-regular array-code ensemble with check degree n. It is also the simplest construction, since degree n check nodes induce degree r checks in the decoding graph. More general constructions in subsequent sub-sections require analysis and manipulation of non-trivial combinatorial coefficients in the induced degree distribution polynomials.

Proposition 5. *The degree distributions given by*

$$\rho(x) = x^{n-1}, \qquad \lambda(x) = 1 - (1-x)^{1/(r-1)} \tag{8}$$

define a probabilistically MDS code ensemble for r failed columns.

Proof: Using the standard formula to calculate the ensemble rate we get

$$\text{rate} = 1 - \frac{\int_0^1 \rho(x)dx}{\int_0^1 \lambda(x)dx} = 1 - \frac{1/n}{1/r} = 1 - \frac{r}{n}.$$

Now for r failed columns the induced degree distributions are

$$\tilde{\rho}(x) = x^{r-1}, \qquad \tilde{\lambda}(x) = 1 - (1-x)^{1/(r-1)}.$$

It is easy to verify that $\tilde{\lambda}(1 - \tilde{\rho}(1-x)) = x$, and hence $x > \epsilon\tilde{\lambda}(1 - \tilde{\rho}(1-x))$ for all $\epsilon < 1$. Hence the ensemble is probabilistically MDS. ■

The array-code ensemble of Proposition 5 has a check-regular degree distribution (all the check nodes have equal degree n), and is a close relative of the check-regular matched distributions used in the construction of capacity-approaching one-dimensional LDPC codes [17]. Unlike in the one-dimensional case, here it is not necessary to modify the distributions (8) to get $\text{rate} > 0$. In the two-dimensional construction, the distributions (8) only have their *induced* rate equal to 0, with their ensemble rate reaching the optimum of $1 - r/n$.

It is now interesting to examine the performance of the aforementioned probabilistically MDS code when the number of failed columns is $s > r$. For that, we replace the r with s in the induced check-degree distribution $\tilde{\rho}(x)$, but leave the original (design parameter) r in the variable-degree distribution $\tilde{\lambda}(x)$. The modified check-degree distribution will be marked $\tilde{\rho}_s(x)$. Now with $\tilde{\rho}_s(x) = x^{s-1}$ and $\tilde{\lambda}(x) = 1 - (1-x)^{1/(r-1)}$ we need to find ϵ_s as the largest ϵ such that $x > \epsilon\tilde{\lambda}(1 - \tilde{\rho}_s(1-x))$ for every $x \in (0, \epsilon]$. Substituting $\tilde{\rho}_s, \tilde{\lambda}$, we get

$$\epsilon_s = \sup\left\{\epsilon : x > \epsilon\left[1 - (1-x)^{\frac{s-1}{r-1}}\right],\ \forall x \in (0,\epsilon]\right\}$$

Theorem 6. *For $s \geqslant r$ column failures, a code constructed from the ensemble of Proposition* 5 *can with high probability recover from any erasure probability $\epsilon < \epsilon_s = (r-1)/(s-1)$.*

Proof: The case $s = r$ was proved in Proposition 5; now assume $s > r$. We need to find the supremum of ϵ such that $x > \epsilon[1 - (1-x)^{\frac{s-1}{r-1}}]$ for every $x \in (0, \epsilon]$. For convenience, we change variables $y := 1 - x$ and $\sigma := (s-1)/(r-1)$. Now the condition becomes

$$1 - y > \epsilon[1 - y^\sigma],\ \forall y \in [1-\epsilon, 1) \tag{9}$$

Let $f(y) = 1 - y - \epsilon(1 - y^\sigma)$. It is easy to verify that f is convex and that $f(1) = 0$. So to get $f(y) > 0$ for $y < 1$ (as required by (9)), it is both sufficient and necessary to have $f'(y) < 0$ for all $0 < y < 1$ (a zero first derivative at some $y_0 < 1$ implies a local minimum and $f(y) < 0$ for all $y_0 < y < 1$). The above gives the condition

$$f'(y) = \sigma\epsilon y^{\sigma-1} - 1 < 0,\ y \in (0,1),$$

or

$$\epsilon < \frac{1}{\sigma} = \frac{r-1}{s-1} \triangleq \epsilon_s.$$

■

Note that induced-rate considerations imply the following upper bound on ϵ_s when s column failures occur:

$$\epsilon_s < 1 - \text{rate}(\tilde{\lambda}(x), \tilde{\rho}_s(x)) = \frac{\int_0^1 \tilde{\rho}_s(x)dx}{\int_0^1 \tilde{\lambda}(x)dx} = \frac{r}{s}. \tag{10}$$

Therefore, since $(r-1)/(s-1) < r/s$ for all $s > r$, the decoding performance of the check-regular array-code ensemble has a lower threshold than what in principle can be achieved given its induced rate. It is an interesting open problem whether there exist probabilistically MDS codes that have optimal thresholds for $s > r$ as well (or if tighter than (10) upper bounds exist).

B. Check-degree $d < n$ probabilistically MDS codes

Before constructively presenting probabilistically MDS codes with regular check-degree $d < n$, we give a result that limits the parameters d, n, r for which such codes are possible.

Proposition 7. *A probabilistically MDS check-regular array-code ensemble must have* $d > n - r + 1$.

Proof: If $d \leqslant n - r + 1$, then $d - 1 \leqslant n - r$, and by (5) we have $\tilde{R}_1 > 0$. This also implies $\tilde{\rho}_1 > 0$, which in turn implies $\tilde{\rho}(0) > 0$. Therefore, substituting $x = 1$ in the right-hand side of (7) gives $\tilde{\lambda}(1 - \tilde{\rho}(0)) < 1 = x$, in violation with the equality condition of Proposition 4. ■

The next theorem shows that for the special case of $r \leqslant 4$, the necessary condition in Proposition 7 is also sufficient for the existence of regular check-degree $d < n$ probabilistically MDS array codes.

Theorem 8. *For any* $r \leqslant 4$*, there exists a probabilistically MDS array-code ensemble with regular check degree* d*, for any* $d > n - r + 1$.

Proof: For a check-regular degree d ensemble the node-perspective degree distribution polynomial is

$$R(x) = x^d.$$

The induced degree distribution, given the parameters n, r, is (from Theorem 2):

$$\tilde{R}(x) = \frac{1}{\binom{n}{d}} \sum_{i=0}^{r} \binom{r}{i} \binom{n-r}{d-i} x^i. \tag{11}$$

From edge perspective, the induced degree-distribution polynomial is

$$\begin{aligned} \tilde{\rho}(x) = \sum_{i=1}^{r} \tilde{\rho}_i x^{i-1} &= \frac{\tilde{R}'(x)}{\tilde{R}'(1)} \\ &= \frac{1}{\binom{n}{d}\tilde{R}'(1)} \sum_{i=1}^{r} i \binom{r}{i} \binom{n-r}{d-i} x^{i-1}. \end{aligned} \tag{12}$$

Note that the condition $d > n - r + 1$ guarantees that $\tilde{\rho}_1 = 0$, as shown to be necessary in Proposition 7. The next step to obtain a probabilistically MDS ensemble is to find a variable-degree distribution $\tilde{\lambda}(x)$ such that

$$\tilde{\lambda}(1 - \tilde{\rho}(1 - x)) = x,$$

which will guarantee correcting any $\epsilon < 1$ fraction of erasures in r columns. The existence of such a degree distribution polynomial $\tilde{\lambda}(x)$ can be established with the aid of the following lemma from [13].

Lemma 9. *[13] For a polynomial* $f(x) = f_1 x + f_2 x^2 + f_3 x^3$ *with* $f(1) = 1$ *and* $\forall i,\ f_i \geqslant 0$*, there exists a polynomial* $g(x)$ *with* $g(1) = 1$ *and* $\forall i,\ g_i \geqslant 0$ *such that* $g(1 - f(1 - x)) = x$ *if*

$$2f'(1)f'''(1) \leqslant f''(1)^2.$$

(f' is the standard derivative of $f(x)$ with respect to x, and similarly f'' and f''' are the second and third derivatives, respectively.)

Translation of Lemma 9's sufficient condition to a condition on the coefficients of $\tilde{\rho}(x)$ gives

Lemma 10. *For a check (induced) degree-distribution polynomial* $\tilde{\rho}(x) = \tilde{\rho}_2 x + \tilde{\rho}_3 x^2 + \tilde{\rho}_4 x^3$ *(*$\tilde{\rho}(1) = 1, \forall i,\ \tilde{\rho}_i \geqslant 0$*), there exists a variable degree distribution polynomial* $\tilde{\lambda}(x)$ *(*$\tilde{\lambda}(1) = 1, \forall i,\ \tilde{\lambda}_i \geqslant 0$*) such that* $\tilde{\lambda}(1 - \tilde{\rho}(1 - x)) = x$ *if*

$$3\tilde{\rho}_2\tilde{\rho}_4 \leqslant \tilde{\rho}_3^2. \tag{13}$$

Proof: Taking $f = \tilde{\rho}$ in Lemma 9 and substituting

$$f'(1) = \tilde{\rho}_2 + 2\tilde{\rho}_3 + 3\tilde{\rho}_4,\ f''(1) = 2\tilde{\rho}_3 + 6\tilde{\rho}_4,\ f'''(1) = 6\tilde{\rho}_4$$

gives the sufficient condition (13). ■

It is now required to prove that the sufficient condition of Lemma 10 is met for any n and d covered by the theorem's assumptions.

Denote $Z = \binom{n}{d}\tilde{R}'(1)$. If $r < 4$, $\tilde{\rho}_4 = 0$ and the sufficient condition (13) is met trivially, therefore we assume $r = 4$. Substituting $r = 4$ in (12) gives

$$Z\tilde{\rho}_2 = 12\binom{n-4}{d-2},\ Z\tilde{\rho}_3 = 12\binom{n-4}{d-3},\ Z\tilde{\rho}_4 = 4\binom{n-4}{d-4}$$

To prove that $\tilde{\rho}_2, \tilde{\rho}_3, \tilde{\rho}_4$ above satisfy the sufficient condition (13), we derive successively simpler equivalent conditions

$$\begin{aligned}
3\tilde{\rho}_2\tilde{\rho}_4 &\leqslant \tilde{\rho}_3^2 \\
&\Updownarrow \\
\frac{144}{Z^2}\binom{n-4}{d-2}\binom{n-4}{d-4} &\leqslant \frac{144}{Z^2}\binom{n-4}{d-3}\binom{n-4}{d-3} \\
&\Updownarrow \\
\frac{(n-d-1)!(n-d-1)!}{(d-2)!(d-4)!} &\leqslant \frac{(n-d-2)!(n-d)!}{(d-3)!(d-3)!} \\
&\Updownarrow \\
(d-3)(n-d-1) &\leqslant (d-2)(n-d) \\
&\Updownarrow \\
nd-d^2-d-3n+3d+3 &\leqslant nd-d^2-2n+2d \\
&\Updownarrow \\
3 &\leqslant n.
\end{aligned}$$

All the transitions follow simple arithmetic operations to both sides of the inequality. Therefore, the condition $n \geqslant 3$ is equivalent to (13), and is sufficient for the existence of the desired variable-degree distribution polynomial $\tilde{\lambda}(x)$. Since codes with fewer than 3 columns are not very interesting, $n \geqslant 3$ is met for every useful d, n parameters.
The final step in proving the probabilistic-MDS property is showing that the rate of the code ensemble equals $1 - r/n$. This fact is established in the following Proposition 11. ■

Proposition 11. *Let $\rho(x) = x^{d-1}$ induce a check degree distribution $\tilde{\rho}(x)$ as in* (12)*. Let $\lambda(x)$ be a variable-degree distribution satisfying $\lambda(1-\tilde{\rho}(1-x)) = x$. Then*

$$1 - \frac{\int_0^1 \rho(x)dx}{\int_0^1 \lambda(x)dx} = 1 - \frac{r}{n}.$$

Proof: We first observe that if $\lambda(1-\tilde{\rho}(1-x)) = x$ then $\int_0^1 \lambda(x)dx = \int_0^1 \tilde{\rho}(x)dx$. This is seen by rewriting the former as $1-\tilde{\rho}(1-x) = \lambda^{-1}(x)$ and the fact that $\int_0^1 \lambda^{-1}(x)dx = 1 - \int_0^1 \lambda(x)dx$. So we are now to prove the equivalent statement

$$\frac{\int_0^1 \rho(x)dx}{\int_0^1 \tilde{\rho}(x)dx} = \frac{r}{n}. \tag{14}$$

The inverses of the numerator and denominator of the left-hand side are, respectively,

$$\begin{aligned}
\frac{1}{\int_0^1 \rho(x)dx} &= d \\
\frac{1}{\int_0^1 \tilde{\rho}(x)dx} &= \frac{1}{\binom{n}{d}}\sum_{i=1}^{r} i\binom{r}{i}\binom{n-r}{d-i},
\end{aligned}$$

where the latter follows from the relation $\int_0^1 \tilde{\rho}(x)dx = 1/\tilde{R}'(1)$ and from (11). We now write

$$\begin{aligned}
\frac{1}{\int_0^1 \tilde{\rho}(x)dx} &= \frac{1}{\binom{n}{d}}\sum_{i=1}^{r} i\binom{r}{i}\binom{n-r}{d-i} \\
&= \frac{r}{\binom{n}{d}}\sum_{i=1}^{r} \frac{i}{r}\binom{r}{i}\binom{n-r}{d-i} \\
&= \frac{r}{\binom{n}{d}}\sum_{i=1}^{r} \binom{r-1}{i-1}\binom{n-r}{d-i} \qquad (15) \\
&= \frac{r}{\binom{n}{d}}\binom{n-1}{d-1} \\
&= r\frac{d}{n}.
\end{aligned}$$

The sum in (15) counts all column combinations such that one column is a fixed failed column, $i-1$ columns are chosen from the remaining $r-1$ failed columns, and $d-i$ columns are chosen from the non-failed columns. Since $i-1$ takes the full range from 0 to $r-1$, (15) sums all combinations of choosing $d-1$ columns from the $n-1$ columns that are not the fixed failed

column. Hence the $\binom{n-1}{d-1}$ in the subsequent expression. All the other transitions are elementary arithmetic manipulations. Now we arrived at the needed result $\int_0^1 \rho(x)dx / \int_0^1 \tilde{\rho}(x)dx = r/n$ ■

At the heart of Proposition 11 lies (14), the fact that the ratio between the average degree of a check node in the code graph $(\int_0^1 \rho(x)dx)^{-1}$, and the average degree of a check node in the induced graph $(\int_0^1 \tilde{\rho}(x)dx)^{-1}$, is n/r (same as the ratio between the number of columns in the code block and the number of failing columns seen by the decoder). This fact is true for general $\rho(x)$, not just regular $\rho(x)$ as above. However, since Theorem 8 applies to regular $\rho(x)$, the simpler proof of the special case suffices.

VI. Comparison with One-Dimensional LDPC Codes

The value of LDPC array codes has been demonstrated in previous sections in two main respects:

1) They correct erasures beyond the designed number of failed columns, unlike traditional array codes (Section IV).
2) Probabilistically MDS codes attain optimal redundancy (Section V).

In this section, it is our objective to show, and theoretically quantify, the value of LDPC array codes in more generality than in previous sections. For this analysis, the erasure-correction performance of LDPC array-code ensembles will be measured using precise theoretical tools. The generality of the forthcoming study is embodied in its applicability to codes of all parameters, not just probabilistically MDS ones.

The most natural way to study the performance of LDPC array codes is through a *comparison with one-dimensional LDPC codes*. It is possible that for the two-dimensional error model defined in Section II, one may choose to use a standard one-dimensional LDPC code, i.e., a graph with bn variable nodes drawn from a degree-distribution pair without the one-neighbor-per-column restriction of Construction 1. Then the key question for the evaluation of LDPC array codes is whether they provably outperform the alternative choice of one-dimensional LDPC codes, and if so, by how much. As shown in the remainder of this section, LDPC array codes are provably better than one-dimensional LDPC codes for all parameters, and their advantage can be quantified analytically.

A. Induced tree ensembles

The decoding performance of codes over the proposed two-dimensional erasure model is determined by the structure of the decoding graph induced by the r failing columns. For that reason, the performance analysis in previous sections has made extensive use of the codes' induced degree distributions. It is apparent that further theoretical understanding of the codes' performance depends on our ability to analyze and manipulate induced degree distributions with general parameters. Unfortunately, the coefficients of the hypergeometric distribution as given in (5) are calculated as multiplications and divisions of different binomial coefficients, which are difficult to deal with analytically. To go around this difficulty, we refine our view of induced degree distributions by examining the *induced tree ensembles* of the codes. Recall [15] that a tree ensemble is the asymptotic version of the decoding-graph ensemble, consisting of rooted bi-partite trees whose degrees are distributed according to some variable/check degree-distribution pair. For notational simplicity, in our discussion on tree ensembles we focus on regular code ensembles with check degree d. However, similar constructs and analysis can be provided for irregular check degrees as well.

The first tree ensemble we examine is the one induced by the regular Construction 1.

Tree Ensemble 1 *The constrained random sampling in Construction* 1 *implies that the d variable nodes connected to a check node reside in d distinct columns, which form a uniformly selected size-d subset of the column set $\{1, \ldots, n\}$. Given a set of r failing columns, the intersection size i between the size-r failing subset and a random size-d subset follows the hypergeometric distribution*

$$P_i^{\square} = \frac{\binom{r}{i}\binom{n-r}{d-i}}{\binom{n}{d}}. \tag{16}$$

Hence (16) *is the induced check-degree distribution of the tree ensemble of Construction* 1. *The superscript $\square$ represents a two-dimensional ensemble.*

As noted earlier, analyzing Tree Ensemble 1 above is difficult due to the combinatorial form of (16). For that reason, we propose Tree Ensemble 2 as a slight variation of Tree Ensemble 1 that is more amenable to analysis.

Tree Ensemble 2 *Instead of choosing the neighboring columns of a check node as size-d subsets of $\{1, \ldots, n\}$, in the modified tree ensemble a check node connects to each column in $\{1, \ldots, n\}$ i.i.d. with probability $\delta = d/n$. The result of this modified tree ensemble is that the intersection size i between the size-r failing subset and the set of neighboring columns in the tree now follows the binomial distribution*

$$P_i^{\boxdot} = \binom{r}{i}\delta^i(1-\delta)^{r-i}. \tag{17}$$

The superscript $\boxdot$ represents a modified two-dimensional ensemble.

Note that in Tree Ensemble 2 the check degrees are d only in expectation, so if we change the graph sampling of Construction 1 to obtain Tree Ensemble 2, the code graph becomes not strictly check-regular. Nevertheless, this minor deviation from regularity does not impose any practical implementation issue.
The last tree ensemble we consider is the one induced by standard one-dimensional LDPC codes drawn with no column restrictions. It is given in Tree Ensemble 3.

Tree Ensemble 3 *In a regular check-degree d one-dimensional LDPC code a check node may connect to two (or more) variable nodes in the same column. In particular, given a size-r subset of columns, each edge connects to a variable node in this subset i.i.d. with probability $\beta = r/n$. As a result, in the induced tree ensemble of one-dimensional codes, the number of edges i connecting to variable nodes in the failing subset follows the binomial distribution*

$$P_i^{-} = \binom{d}{i} \beta^i (1-\beta)^{d-i}. \tag{18}$$

The superscript $-$ represents a one-dimensional ensemble.

Note that both Tree Ensemble 2 and Tree Ensemble 3 result in binomial check-degree distributions in the induced graphs, but with different parameters. The binomial distributions of (17) and (18) are well-known approximations [6] for the hypergeometric distribution (16). In the remainder of the section, our objective is to compare the 2D Tree Ensemble 2 and the 1D Tree Ensemble 3 in terms of their iterative-decoding performance.

B. Iterative-decoding analysis of induced tree ensembles

As we replaced the unwieldy hypergeometric distribution $P_i^{\square}$ of (16) with the more manageable binomial distribution $P_i^{\boxdot}$ in (17), now we can express the induced check-degree distribution polynomial (from node perspective) of LDPC array codes as

$$\tilde{R}^{\boxdot}(x) = \sum_{i=0}^{r} \binom{r}{i} \delta^i (1-\delta)^{r-i} x^i = (1-\delta+\delta x)^r. \tag{19}$$

In a similar way, the induced check-degree distribution polynomial of one-dimensional LDPC codes is

$$\tilde{R}^{-}(x) = \sum_{i=0}^{d} \binom{d}{i} \beta^i (1-\beta)^{d-i} x^i = (1-\beta+\beta x)^d. \tag{20}$$

Let $a_{\tilde{R}^{\boxdot}}$ be the average induced check degree in Tree Ensemble 2, and $a_{\tilde{R}^{-}}$ be the average induced check degree in Tree Ensemble 3. A first attempt to differentiate between Tree Ensemble 2 and Tree Ensemble 3 – based on their average induced check degree – turns unsuccessful, as shown in the following proposition.

Proposition 12. *Tree Ensemble 2 and Tree Ensemble 3 have the same average induced check degree*

$$a_{\tilde{R}^{\boxdot}} = a_{\tilde{R}^{-}} = \frac{rd}{n}.$$

Proof: By definition we have $a_{\tilde{R}^{\boxdot}} = \tilde{R}'^{\boxdot}(1)$ and $a_{\tilde{R}^{-}} = \tilde{R}'^{-}(1)$. Taking the derivative of (19) and (20) with respect to x and substituting $x = 1$, $\beta = r/n$, $\delta = d/n$, we get

$$\tilde{R}'^{\boxdot}(1) = \tilde{R}'^{-}(1) = \frac{rd}{n}.$$

■

The fact that the 2D and 1D induced tree ensembles have the same average degree means that we cannot use degree-based arguments to show a performance gap in favor of the 2D construction. A more refined quantitative differentiation between the 1D and 2D induced degree distributions is given in the following proposition.

Proposition 13. *Given code parameters d,n, and $r < d$ failed columns, the degree distributions of the induced 1D and 2D tree ensembles satisfy*

$$\tilde{R}^{\boxdot}(x) < \tilde{R}^{-}(x),$$

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

Proof: We substitute $y = 1 - x$ in the degree distribution polynomials at the right-hand side of (19) and (20); then use the following lemma.

Lemma 14. *For any $\delta > \beta$, $\delta, \beta \in (0,1]$*

$$(1-\delta y)^{\beta} < (1-\beta y)^{\delta},$$

for all $y \in (0,1]$.

Proof: We expand the two sides of the inequality using fractional binomial coefficients.

$$(1-\delta y)^{\beta} = 1 - \binom{\beta}{1}\delta y + \binom{\beta}{2}\delta^2 y^2 - \binom{\beta}{3}\delta^3 y^3 + \cdots \tag{21}$$

and

$$(1-\beta y)^{\delta} = 1 - \binom{\delta}{1}\beta y + \binom{\delta}{2}\beta^2 y^2 - \binom{\delta}{3}\beta^3 y^3 + \cdots \tag{22}$$

The expansion of fractional binomial coefficient is given by

$$\begin{aligned}\binom{\alpha}{\ell} &= \frac{\alpha(\alpha-1)\cdots(\alpha-\ell+1)}{\ell!} \\ &= (-1)^{\ell-1}\frac{\alpha}{\ell}\left(1-\frac{\alpha}{\ell-1}\right)\cdots\left(1-\frac{\alpha}{2}\right)(1-\alpha).\end{aligned}$$

A simple observation is that for $\delta > \beta$, we have $\frac{1}{\beta}\binom{\beta}{\ell} > \frac{1}{\delta}\binom{\delta}{\ell}$ for odd ℓ, and $\frac{1}{\beta}\binom{\beta}{\ell} < \frac{1}{\delta}\binom{\delta}{\ell}$ for even ℓ. As a result, we have for odd ℓ

$$\binom{\beta}{\ell}\delta^{\ell} = \frac{1}{\beta}\binom{\beta}{\ell}\delta^{\ell}\beta > \frac{1}{\delta}\binom{\delta}{\ell}\delta^{\ell}\beta > \binom{\delta}{\ell}\beta^{\ell},$$

where the last inequality follows directly from $\delta > \beta$. This proves that the coefficients of odd powers of y are smaller in (21) than in (22). In a similar way we have for even ℓ

$$\binom{\beta}{\ell}\delta^{\ell} < \binom{\delta}{\ell}\beta^{\ell},$$

which proves that the coefficients of non-zero even powers of y are smaller in (21) than in (22). This proves that

$$(1-\delta y)^{\beta} < (1-\beta y)^{\delta}.$$

■

To prove the proposition, we recall that

$$\tilde{R}^{\boxdot}(x) = \tilde{R}^{\boxdot}(1-y) = (1-\delta y)^{\beta n}$$

and

$$\tilde{R}^{-}(x) = \tilde{R}^{-}(1-y) = (1-\beta y)^{\delta n},$$

so $(1-\delta y)^{\beta} < (1-\beta y)^{\delta}$ from Lemma 14 implies $\tilde{R}^{\boxdot}(x) < \tilde{R}^{-}(x)$. ■

With the proven gap between $\tilde{R}^{\boxdot}(x)$ and $\tilde{R}^{-}(x)$ given by Proposition 13, we are now ready to prove the main result of this section.

Theorem 15. *For any code parameters n, d, and a number of failed columns $r < d$, the rate of a 1D check-regular ensemble is bounded strictly below the rate of a 2D ensemble for the same erasure probability ϵ.*

Proof: To prove a gap in the maximal rate between 1D and 2D codes we use a known result from the theory of one-dimensional LDPC codes.

Lemma 16. *[17] For a code ensemble with check-degree distribution polynomial $R(x)$, average check-node degree a_R, and average variable-node degree a_L, successful BP decoding is possible if the erasure probability ϵ satisfies*

$$\epsilon < \frac{a_L}{a_R}(1 - R(1-\epsilon)).$$

Substituting the parameters of the induced decoding graphs into Lemma 16, we get two upper bounds. One for the 1D code

$$\epsilon < \frac{a_{L-}}{a_{\tilde{R}-}}(1 - \tilde{R}^{-}(1-\epsilon)),$$

and one for the 2D code

$$\epsilon < \frac{a_{L\boxdot}}{a_{\tilde{R}\boxdot}}(1 - \tilde{R}^{\boxdot}(1-\epsilon)).$$

Substituting $a_{\tilde{R}\boxdot} = a_{\tilde{R}-} = \frac{rd}{n}$ and rearranging, we get

$$a_{L-} > \frac{rd\epsilon}{n(1-\tilde{R}^{-}(1-\epsilon))},$$

and

$$a_{L\boxdot} > \frac{rd\epsilon}{n(1-\tilde{R}^{\boxdot}(1-\epsilon))}.$$

The average check degree in the *code* graph is the same for the 1D and 2D construction

$$a_{R-} = a_{R\boxdot} = d.$$

As a result, the lower bounds on a_{L-} and $a_{L\boxdot}$ readily translate to upper bounds on the respective code rates through the relation $\mathrm{rate} = 1 - \frac{a_L}{a_R}$.

$$\mathrm{rate}_{1\mathrm{D}} < 1 - \frac{r\epsilon}{n(1-\tilde{R}^{-}(1-\epsilon))}, \tag{23}$$

and

$$\mathrm{rate}_{2\mathrm{D}} < 1 - \frac{r\epsilon}{n(1-\tilde{R}^{\boxdot}(1-\epsilon))}. \tag{24}$$

Having proved in Proposition 13 that $\tilde{R}^{\boxdot}(1-\epsilon) < \tilde{R}^{-}(1-\epsilon)$, we establish that the right-hand side of (23) is strictly smaller than that of (24). A gap between the allowable rates of 1D and 2D codes is now proven. ■

The implication from Theorem 15 is that for the two-dimensional r, ϵ erasure model proposed in Section II, two-dimensional codes may give better rates than standard one-dimensional codes that ignore the structure of the array. While the above only shows a gap in the upper bound on the rates, it is possible to construct explicit 2D ensembles with rates that exceed the upper bound for 1D codes. For example, the special case $d = n$ ($\delta = 1$), $\epsilon = 1$, for which we have shown codes with $\mathrm{rate}_{2\mathrm{D}} = 1 - r/n$ (Proposition 5), has a strictly smaller rate upper bound for 1D codes.

It is important to note that our re-definition of 2D codes as Tree Ensemble 2 instead of Tree Ensemble 1 was done solely to gain analytic tractability in proving results such as Theorem 15. In fact, in all the parameters that we checked, we have found that $\tilde{R}^{\square}(x) < \tilde{R}^{\boxdot}(x) < \tilde{R}^{-}(x)$ for all $x \in [0,1)$, hence the original 2D sampling (Construction 1) giving rise to Tree Ensemble 1 is expected to give even better rates than Tree Ensemble 2 analyzed in this section.

C. Experimental validation

In addition to the theoretical advantage of 2D codes proved in the previous sub-section, we now want to see an example of this advantage on a real code. To get such as example, we took an array with dimensions $n = 16$, $b = 1000$. With fixed variable-node degree $l = 3$ and fixed check-node degree $d = 12$, we randomly drew a 2D code according to Construction 1, and a 1D code similarly, only without the column restrictions. For the case of $r = 6$ failing columns, we simulated i.i.d. erasures within failing columns with varying erasure probabilities ϵ. For each of the codes we measured the decoding-success percentage across many channel instances, each randomly choosing the r failing columns and the erasures within columns. Decoding success is defined as recovering the entire erased bits after a fixed number of iterations. The results are given in Figure 6. It can be seen that the 2D code outperforms the 1D code in many ϵ points, never performing worse. The theoretical 2D threshold for $n = 16, l = 3, d = 12, r = 6$ is marked as a vertical solid line at $\epsilon = 0.56$.

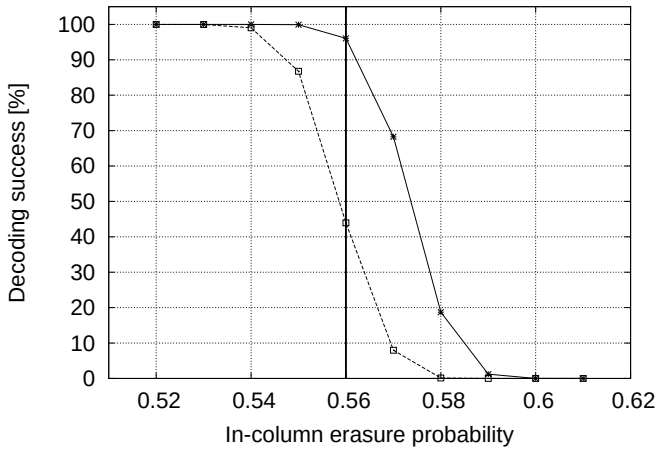

Figure 6. Experimental comparison between decoding success of a 2D code (star markers) and a 1D code (square markers). For each in-column erasure probability ϵ the plots mark the percentage of decoding instances that resulted in full recovery of erased bits.

VII. Conclusion

The results of the paper serve to lay out a new theoretical framework for LDPC codes over two-dimensional arrays. From these results many open questions arise. The main open problem on the constructive side is the search for code ensembles that give optimal decoding performance for multiple r values. More upper bounds on code rates given correction parameters are also important to come by. From practical perspective, it is beneficial to consider similar codes that are systematic.

VIII. Acknowledgement

We would like to thank Uri Gertzek and Gali Granot for the implementation and simulations that led to the results of Figure 6. We also want to thank Igal Sason for valuable discussions.

References

[1] M. Blaum, J. Bruck, and A. Vardy, "MDS array codes with independent parity symbols," *IEEE Transactions on Information Theory*, vol. 42, no. 2, pp. 529–542, 1996.
[2] M. Blaum, J. L. Fan, and L. Xu, "Soft decoding of several classes of array codes," in *Proc. of the IEEE International Symposium on Info. Theory*, Lausanne, Switzerland, June 2002, p. 368.
[3] M. Blaum, P. Farrell, and H. van Tilborg, "Array codes," *Handbook of Coding Theory, V.S. Pless and W.C. Huffman*, pp. 1855–1909, 1998.
[4] M. Blaum, J. Hafner, and S. Hetzler, "Partial-MDS codes and their application to RAID type of architectures," Arxiv.org, to appear in IEEE Transactions on Information Theory, http://arxiv.org/abs/1205.0997, Tech. Rep. cs.IT/1205.0997, 2012.
[5] M. Blaum and R. Roth, "New array codes for multiple phased burst correction," *IEEE Transactions on Information Theory*, vol. 39, no. 1, pp. 66–77, 1993.
[6] H. Brunk, J. Holstein, and F. Williams, "A comparison of binomial approximations to the hypergeometric distribution," *The American Statistician*, vol. 22, no. 1, pp. 24–26, 1968.
[7] Y. Cassuto and J. Bruck, "Low-complexity array codes for random and clustered 4-erasures," *IEEE Transactions on Information Theory*, vol. 58, no. 1, pp. 146–158, 2012.
[8] Y. Cassuto and A. Shokrollahi, "Array-code ensembles -or- two-dimensional LDPC codes," in *Proc. of the IEEE International Symposium on Info. Theory*, St. Petersburg, Russia, July 2011, pp. 518–522.
[9] J. L. Fan, "Array codes as low-density parity check codes," in *Proc. of the Intl. Symp. on Turbo Codes*, 2000, pp. 543–546.
[10] W. Feller, *An Introduction to Probability Theory and Its Applications Volume I, third edition*. Wiley, 1968.
[11] M. Luby, M. Mitzenmacher, and A. Shokrollahi, "Analysis of random processes via and-or trees," in *Proc. of the 9th Annual ACM SIAM Symposium on Discrete Algorithms*, 1998, pp. 364–373.
[12] M. Luby, M. Mitzenmacher, A. Shokrollahi, and D. Spielman, "Efficient erasure correcting codes," *IEEE Transactions on Information Theory*, vol. 47, no. 2, pp. 569–584, 2001.
[13] P. Oswald and A. Shokrollahi, "Capacity-achieving sequences for the erasure channel," *IEEE Transactions on Information Theory*, vol. 48, no. 12, pp. 3017–3028, 2002.
[14] D. A. Patterson, G. A. Gibson, and R. Katz, "A case for redundant arrays of inexpensive disks," in *Proc. SIGMOD Int. Conf. Data Management*, 1988, pp. 109–116.
[15] T. Richardson and R. Urbanke, *Modern coding theory*. New York USA: Cambridge University Press, 2008.
[16] R. Roth and P. Vontobel, "Coding for combined block-symbol error correction," in *Proc. of the IEEE International Symposium on Info. Theory*, Istanbul, Turkey, July 2013, pp. 1217–1221.
[17] A. Shokrollahi, "New sequences of linear time erasure codes approaching the channel capacity," in *Proc. 13th International Symposium on Applied Algebra, Algebraic Algorithms, and Error-Correcting Codes. Lecture Notes in Computer Science 1719*, 1999, pp. 65–76.