On the Minimum Distance of Array-Based Spatially-Coupled Low-Density Parity-Check Codes

Eirik Rosnes

Department of Informatics, University of Bergen, N-5020 Bergen, Norway, and the Simula Research Lab.
Email: eirik@ii.uib.no

Abstract—An array low-density parity-check (LDPC) code is a quasi-cyclic LDPC code specified by two integers q and m, where q is an odd prime and $m \leq q$. The exact minimum distance, for small q and m, has been calculated, and tight upper bounds on it for $m \leq 7$ have been derived. In this work, we study the minimum distance of the spatially-coupled version of these codes. In particular, several tight upper bounds on the *optimal* minimum distance for coupling length at least two and $m = 3, 4, 5$, that are independent of q and that are valid for all values of $q \geq q_0$ where q_0 depends on m, are presented. Furthermore, we show by exhaustive search that by carefully selecting the *edge spreading* or *unwrapping* procedure, the minimum distance (when q is not very large) can be significantly increased, especially for $m = 5$.

I. INTRODUCTION

In this paper, we consider array-based spatially-coupled low-density parity-check (SC-LDPC) codes as introduced in [1] and subsequently studied in [2] from the perspective of *absorbing sets*. Array LDPC codes were originally proposed by Fan in [3] and are specified by two integers q and m, where q is an odd prime and $m \leq q$.

Spatial coupling was first introduced in the coding theory literature by Felström and Zigangirov in [4], where they proposed convolutional LDPC or SC-LDPC codes. These codes have very good belief propagation (BP) decoding thresholds due to the phenomenon of *threshold saturation*. For instance, in [5], it was shown that for binary memoryless channels, the BP decoding threshold *saturates* to the maximum *a posteriori* decoding threshold of the underlying ensemble.

Since the original work by Fan [3], several authors have considered the *structural* properties (including the minimum distance $d_{\min}$) of array LDPC codes (see, e.g., [6–11]). For high rate and moderate length, these codes perform well under iterative decoding, and they are also well-suited for practical implementation due to their regular structure [12, 13].

In this work, we present several tight upper bounds on the *optimal* minimum distance of array-based SC-LDPC codes for coupling length at least two and $m = 3, 4, 5$, that are independent of q and that are valid for all values of $q \geq q_0$ where q_0 depends on m. Also, these bounds (and intermediate results in their proofs) can in some cases be used to quickly remove large parts of the search space when searching for *optimal cutting vectors*. For small values of q ($m = 3, 4, 5$)

we present the results of an exhaustive search over all cutting vectors, showing that a careful selection can increase the minimum distance significantly, especially for $m = 5$.

We remark that the iterative decoding performance of these codes, for instance, on additive white Gaussian noise channels, is typically dominated by minimum absorbing sets and their multiplicities. However, improved decoding methods and ultimately maximum-likelihood decoding will not be trapped in absorbing sets and thus overcome the shortcomings of standard iterative decoding, in which case the minimum distance (and its multiplicity) will become an important performance metric.

II. ARRAY LDPC CODES

The array LDPC code $\mathcal{C}(q, m)$, with parameters q and m, has length q^2 and can be defined by the parity-check matrix

$$\boldsymbol{H}(q, m) = \begin{bmatrix} \boldsymbol{I} & \boldsymbol{I} & \boldsymbol{I} & \cdots & \boldsymbol{I} \\ \boldsymbol{I} & \boldsymbol{P} & \boldsymbol{P}^2 & \cdots & \boldsymbol{P}^{q-1} \\ \boldsymbol{I} & \boldsymbol{P}^2 & \boldsymbol{P}^4 & \cdots & \boldsymbol{P}^{2(q-1)} \\ & & \vdots & & \vdots \\ \boldsymbol{I} & \boldsymbol{P}^{m-1} & \boldsymbol{P}^{2(m-1)} & \cdots & \boldsymbol{P}^{(m-1)(q-1)} \end{bmatrix} \tag{1}$$

where $\boldsymbol{I}$ is the $q \times q$ identity matrix and $\boldsymbol{P}$ the $q \times q$ permutation matrix

$$\boldsymbol{P} = \begin{bmatrix} 0 & 0 & \cdots & 0 & 1 \\ 1 & 0 & \cdots & 0 & 0 \\ 0 & 1 & \cdots & 0 & 0 \\ & \vdots & & \vdots & \\ 0 & 0 & \cdots & 1 & 0 \end{bmatrix}.$$

Since the number of ones in each row of the matrix in (1) is q and the number of ones in each column is m, the array LDPC codes are (m, q)-regular codes. Furthermore, its rank is $qm - m + 1$, from which it follows that the dimension of $\mathcal{C}(q, m)$ is $q^2 - qm + m - 1$.

In [7], a new representation for $\boldsymbol{H}(q, m)$ was introduced. In particular, since each column of the parity-check matrix $\boldsymbol{H}(q, m)$ has m blocks and each block is a permutation of $(1, 0, 0, \ldots, 0, 0)^T$, where $(\cdot)^T$ denotes the transpose of its argument, we can represent each column as a vector of integers between 0 and $q - 1$, where

$$i \triangleq \left(\overbrace{0, \ldots, 0}^{i}, 1, \overbrace{0, \ldots, 0}^{q-i-1} \right)^T \tag{2}$$

i.e., the 1-positions are associated with the integers modulo q. Furthermore, it follows from (1) and the integer representation in (2) that any column in an array LDPC code parity-check matrix is of the form

$$(x, x + y, x + 2y, \ldots, x + (m - 1)y)^T \bmod q \qquad (3)$$

where $(x, y) \in [q]^2$ and, for notational convenience, for any positive integer L, $[L] \triangleq \{0, 1, \ldots, L - 1\}$. Thus, a column can be specified by two integers x and y. Also, note that since there are q^2 distinct columns in an array LDPC code parity-check matrix, any pair $(x, y) \in [q]^2$ specifies a valid column.

In the following, it is also convenient to consider the parity-check matrix in (1) as an $m \times q$ array of $q \times q$ permutation matrices with *row group* indices $i \in [m]$ and *column group* indices $j \in [q]$, from which it follows that each column can be uniquely indexed by the pair (j, k), where $j \in [q]$ is the index of the column group and $k \in [q]$ is the index within a column group. Likewise, each row can be uniquely indexed by the pair (i, k), where $i \in [m]$ is the index of the row group and $k \in [q]$ is the index within a row group.

III. ARRAY-BASED SC-LDPC CODES

Array-based SC-LDPC codes can be constructed from array LDPC codes by a special type of *edge spreading* or *unwrapping* procedure [4, 14] specified by a "cutting" vector $\boldsymbol{\zeta} = (\zeta_0, \ldots, \zeta_{m-1})$, where $0 \leq \zeta_0 < \zeta_1 < \cdots < \zeta_{m-1} \leq q$. The purpose of the cutting vector is to generate two parity-check matrices $\boldsymbol{H}_0$ and $\boldsymbol{H}_1$, each of size $mq \times q^2$ and initially filled with zeros, as follows:

- For each ζ_i, $i \in [m]$, the $q \times q$ permutation matrices in row group i and column group j, $j < \zeta_i$, of $\boldsymbol{H}(q, m)$ are copied into the corresponding positions in $\boldsymbol{H}_0$.
- Similarly, for each ζ_i, $i \in [m]$, the $q \times q$ permutation matrices in row group i and column group j, $\zeta_i \leq j < q$, of $\boldsymbol{H}(q, m)$ are copied into the corresponding positions in $\boldsymbol{H}_1$.

Example 1: If $q = 5$, $m = 3$, and $\boldsymbol{\zeta} = (1, 2, 4)$, the matrices $\boldsymbol{H}_0$ and $\boldsymbol{H}_1$ become

$$\boldsymbol{H}_0 = \begin{bmatrix} \boldsymbol{I} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{I} & \boldsymbol{P} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{I} & \boldsymbol{P}^2 & \boldsymbol{P}^4 & \boldsymbol{P} & \boldsymbol{0} \end{bmatrix}$$

and

$$\boldsymbol{H}_1 = \begin{bmatrix} \boldsymbol{0} & \boldsymbol{I} & \boldsymbol{I} & \boldsymbol{I} & \boldsymbol{I} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{P}^2 & \boldsymbol{P}^3 & \boldsymbol{P}^4 \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{P}^3 \end{bmatrix}.$$

For a given positive integer L (the coupling length) and a cutting vector $\boldsymbol{\zeta}$, an array-based SC-LDPC code is defined by the parity-check matrix

$$\boldsymbol{H}(q, m, L, \boldsymbol{\zeta}) = \begin{bmatrix} \boldsymbol{H}_0 & & & \\ \boldsymbol{H}_1 & \boldsymbol{H}_0 & & \\ & \boldsymbol{H}_1 & \ddots & \\ & & \ddots & \boldsymbol{H}_0 \\ & & & \boldsymbol{H}_1 \end{bmatrix} \qquad (4)$$

of size $(L + 1)mq \times Lq^2$.

As for the uncoupled case, it is convenient to consider the convolutional parity-check matrix in (4) as a $(L + 1)m \times Lq$ array of $q \times q$ permutation and all-zero matrices in which mq consecutive rows (resp. q^2 consecutive columns) are referred to as a row (resp. column) *section*. Within each row (resp. column) section, the rows (resp. columns) correspond to a row (resp. column) group. In summary, each column can be indexed by a triple (l, j, k), where $l \in [L]$ is the column section index, $j \in [q]$ is the column group index within column section l, and $k \in [q]$ is the column index within column group j in column section l. Similarly, each row can be indexed by a triple (l, i, k), where $l \in [L + 1]$ is the row section index, $i \in [m]$ is the row group index within row section l, and $k \in [q]$ is the row index within row group i in row section l.

Note that as for the uncoupled matrix in (1), all columns in $\boldsymbol{H}(q, m, L, \boldsymbol{\zeta})$ contain exactly m ones and each row in row sections $1, 2, \ldots, L - 1$ contains exactly q ones. However, the rows in row sections 0 and L can have fewer than q ones.

In the following, we will denote the binary linear code defined by the coupled parity-check matrix $\boldsymbol{H}(q, m, L, \boldsymbol{\zeta})$ in (4) by $\mathcal{C}(q, m, L, \boldsymbol{\zeta})$ and its corresponding minimum (resp. stopping) distance by $d(q, m, L, \boldsymbol{\zeta})$ (resp. $h(q, m, L, \boldsymbol{\zeta})$).

It was shown in [2, Theorem 5] that the minimum distance $d(q, m, L, \boldsymbol{\zeta})$ of $\mathcal{C}(q, m, L, \boldsymbol{\zeta})$ is bounded below by the minimum distance $d(q, m)$ of $\mathcal{C}(q, m)$ for all cutting vectors $\boldsymbol{\zeta}$.

Now, let us define the optimal minimum distance of an array-based SC-LDPC code as the highest minimum distance over all possible cutting vectors as the coupling length L tends to infinity, i.e.,

$$d_{\mathrm{opt}}(q, m) \triangleq \lim_{L \to \infty} d_{\mathrm{opt}}(q, m, L)$$

where

$$d_{\mathrm{opt}}(q, m, L) \triangleq \max_{\boldsymbol{\zeta}} d(q, m, L, \boldsymbol{\zeta}).$$

IV. UPPER BOUNDS ON $d_{\mathrm{opt}}(q, m, L)$

In this section, we derive upper bounds on $d_{\mathrm{opt}}(q, m, L)$ for $L \geq 2$ and $m = 3, 4, 5$, that are independent of q and that hold for all values of $q \geq q_0$ where q_0 depends on m.

Now, define the *range* of a sorted (in nondecreasing order) sequence of column group indices $\boldsymbol{j}^{\mathrm{gr}} = (j_0^{\mathrm{gr}}, \ldots, j_{n-1}^{\mathrm{gr}})$ of length n as

$$\min_{l \in [n]} \left(j_l^{\mathrm{gr}} - \left(j_{(l+1) \bmod n}^{\mathrm{gr}} - q \left\lceil \frac{n - 1 - l}{n - 1} \right\rceil \right) \right) + 1.$$

Furthermore, define

$$\epsilon(\boldsymbol{\zeta}) \triangleq \max_{j \in [m-1]} \left(\left(\zeta_{(j+1) \bmod m} - \zeta_j \right) \bmod q \right)$$

$$j_{\max}(\boldsymbol{\zeta}) \triangleq \arg \max_{j \in [m-1]} \left(\left(\zeta_{(j+1) \bmod m} - \zeta_j \right) \bmod q \right)$$

for a given cutting vector $\boldsymbol{\zeta}$.

Example 2: Let $n = 4$, $q = 7$, and $\boldsymbol{j}^{\mathrm{gr}} = (0, 1, 2, 5)$. Then, the range of $\boldsymbol{j}^{\mathrm{gr}}$ is

$$\min \left(0 - (1 - 7 \cdot 1), 1 - (2 - 7 \cdot 1), 2 - (5 - 7 \cdot 1), \right.$$
$$\left. 5 - (0 - 7 \cdot 0) \right) + 1 = \min(6, 6, 4, 5) + 1 = 5.$$

A. The Case $m = 3$

Lemma 1: The minimum distance $d_{\mathrm{opt}}(q, 3, L)$, for $L \geq 2$ and $q \geq 13$, is upper-bounded by 6.

Proof: The proof is based on the template codeword from [7, Theorem 4]. For convenience of the reader, we restate the corresponding *template support matrix* below

$$\begin{bmatrix} 0 & 0 & 2i-2k & 2i-2k & -2i & -2i \\ 0 & -2i+k & 0 & -i & -i & -2i+k \\ 0 & -4i+2k & -2i+2k & -4i+2k & 0 & -2i+2k \end{bmatrix} \quad (5)$$

where $q \geq 5$, $i \in [q] \setminus \{0\}$, and $k \in [q]$ with $k \neq i, 2i$, and where all operations are taken modulo q. The template support matrix is obtained by extracting the columns of the parity-check matrix corresponding to the support set of the underlying codeword. When constructing this matrix, the integer representation of the columns from (3) is used.

Now, a column of the general form $(x, x+y, x+2y)^T \bmod q$, $x, y \in [q]$, in this matrix has column group index of y. Thus, the sequence of column group indices corresponding to the template matrix in (5) is

$$(0, -2i+k, -2i+2k, -3i+2k, i, k) \bmod q.$$

For $i = 1$ and $k = 0$, we get the following sorted sequence

$$(-3, -2, -2, 0, 0, 1) \bmod q. \quad (6)$$

Since array LDPC codes are quasi-cyclic with period q, *cyclically incrementing* a sequence of column group indices $\boldsymbol{j}^{\mathrm{gr}} = (j_0^{\mathrm{gr}}, \ldots, j_{n-1}^{\mathrm{gr}})$ of length n of a codeword in an array LDPC code by an integer $\kappa \geq 1$ results in the valid sequence $(j_0^{\mathrm{gr}} + \kappa, \ldots, j_{n-1}^{\mathrm{gr}} + \kappa) \bmod q$ of column group indices of a codeword that is obtained by cyclically shifting the given codeword by κq positions to the right.

If the range of a sorted (in nondecreasing order) sequence of column group indices corresponding to a codeword in the uncoupled array LDPC code is at most $\epsilon(\boldsymbol{\zeta})$, it can always be cyclically incremented so that the corresponding codeword has coordinates with column group index j within $\left[\zeta_{j_{\max}(\boldsymbol{\zeta})}, \zeta_{j_{\max}(\boldsymbol{\zeta})+1}\right)$ when $j_{\max}(\boldsymbol{\zeta}) < m - 1 = 2$, and within $[\zeta_{m-1}, q) \cup [0, \zeta_0)$ when $j_{\max}(\boldsymbol{\zeta}) = m - 1 = 2$. Consequently, there will exist a codeword in the coupled code (for all $L \geq 2$) with all coordinates within the same column section when $j_{\max}(\boldsymbol{\zeta}) < m-1 = 2$, or within two consecutive column sections when $j_{\max}(\boldsymbol{\zeta}) = m - 1 = 2$. Thus, since $\min_{\boldsymbol{\zeta}} \epsilon(\boldsymbol{\zeta}) = \left\lceil \frac{q}{m} \right\rceil = \left\lceil \frac{q}{3} \right\rceil$, there will exist a codeword in the SC-LDPC code of weight 6 for all cutting vectors $\boldsymbol{\zeta}$ as long as

$$\left\lceil \frac{q}{m} \right\rceil = \left\lceil \frac{q}{3} \right\rceil \geq \mathrm{range}\left((-3, -2, -2, 0, 0, 1) \bmod q\right).$$

Since $\mathrm{range}((-3, -2, -2, 0, 0, 1) \bmod q) = 5$ for $q \geq 7$ and 4 for $q = 5$, the smallest q that satisfies this inequality is $q = 13$, and the result follows since the range of (6) for $q \geq 13$ is constant. ∎

Example 3: Consider the case of $q = 7$, $m = 3$, and the template support matrix in (5) for $i = 1$ and $k = 0$. The sorted (in nondecreasing order) sequence of column group indices is $(0, 0, 1, 4, 5, 5)$ which has range 5. By cyclically

incrementing this sequence by two, we get $(0, 0, 2, 2, 3, 6)$. The corresponding support matrices (both of which correspond to codewords) are

$$\begin{bmatrix} 0 & 0 & 2 & 2 & 5 & 5 \\ 0 & 5 & 0 & 6 & 6 & 5 \\ 0 & 3 & 5 & 3 & 0 & 5 \end{bmatrix} \quad \text{and} \quad \begin{bmatrix} 0 & 0 & 2 & 2 & 5 & 5 \\ 2 & 0 & 2 & 1 & 1 & 0 \\ 4 & 0 & 2 & 0 & 4 & 2 \end{bmatrix}$$

respectively. For the cutting vector $\boldsymbol{\zeta} = (4, 5, 6)$, we have $j_{\max}(\boldsymbol{\zeta}) = m - 1 = 2$, $\epsilon(\boldsymbol{\zeta}) = 5$, $[\zeta_{m-1}, q) \cup [0, \zeta_0) = [6, 7) \cup [0, 4) = \{0, 1, 2, 3, 6\}$, and

$$\begin{bmatrix} \boldsymbol{H}_0 & & \\ \boldsymbol{H}_1 & \boldsymbol{H}_0 & \\ & & \boldsymbol{H}_1 \end{bmatrix} = \left[\begin{array}{ccccccc|ccccccc|ccccccc} \boldsymbol{I} & \boldsymbol{I} & \boldsymbol{I} & \boldsymbol{I} & 0 & 0 & 0 & & & & & & & \\ \boldsymbol{I} & \boldsymbol{P} & \boldsymbol{P}^2 & \boldsymbol{P}^3 & \boldsymbol{P}^4 & 0 & 0 & & & & & & & \\ \boldsymbol{I} & \boldsymbol{P}^2 & \boldsymbol{P}^4 & \boldsymbol{P}^6 & \boldsymbol{P} & \boldsymbol{P}^3 & 0 & & & & & & & \\ \hline 0 & 0 & 0 & 0 & \boldsymbol{I} & \boldsymbol{I} & \boldsymbol{I} & \boldsymbol{I} & \boldsymbol{I} & \boldsymbol{I} & \boldsymbol{I} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \boldsymbol{P}^5 & \boldsymbol{P}^6 & \boldsymbol{I} & \boldsymbol{P} & \boldsymbol{P}^2 & \boldsymbol{P}^3 & \boldsymbol{P}^4 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \boldsymbol{P}^5 & \boldsymbol{I} & \boldsymbol{P}^2 & \boldsymbol{P}^4 & \boldsymbol{P}^6 & \boldsymbol{P} & \boldsymbol{P}^3 & 0 \\ \hline & & & & & & & 0 & 0 & 0 & 0 & \boldsymbol{I} & \boldsymbol{I} & \boldsymbol{I} \\ & & & & & & & 0 & 0 & 0 & 0 & 0 & \boldsymbol{P}^5 & \boldsymbol{P}^6 \\ & & & & & & & 0 & 0 & 0 & 0 & 0 & 0 & \boldsymbol{P}^5 \end{array} \right].$$

From the highlighted blue columns in the matrix above we can identify a weight-6 codeword corresponding to the sequence $(0, 0, 2, 2, 3, 6)$ of column group indices mentioned above. For other cutting vectors like $\boldsymbol{\zeta} = (3, 4, 6)$, we get $\epsilon(\boldsymbol{\zeta}) = 4$, which is less than the range, and there will be no codewords of the type in (5) with $i = 1$ and $k = 0$ (there are for $i = 2$ and $k = 0$) in the coupled code.

By combining Lemma 1 with [2, Corollary 6], we get the following theorem.

Theorem 1: For $L \geq 2$ and $q \geq 13$, the minimum distance $d_{\mathrm{opt}}(q, 3, L) = 6$.

B. The Case $m = 4$

Lemma 2: The minimum distance $d_{\mathrm{opt}}(q, 4, L)$, for $L \geq 2$ and $q \geq 41$, is upper-bounded by 10.

Proof: The proof follows the same main idea of the proof of Lemma 1 above using the template support matrix ($q \geq 11$)

$$\begin{bmatrix} 0 & 0 & -16 & -16 & -13 & -13 & -9 & -9 & -1 & -1 \\ 0 & -2 & -10 & -8 & -10 & -6 & -8 & -6 & 0 & -2 \\ 0 & -4 & -4 & 0 & -7 & 1 & -7 & -3 & 1 & -3 \\ 0 & -6 & 2 & 8 & -4 & 8 & -6 & 0 & 2 & -4 \end{bmatrix}$$

which we have found using the algorithm from [10]. The remaining technical details are omitted for brevity. ∎

Note that the template support matrix for $m = 4$ given in Fig. 3 in [8] will fail to prove the above result, since the sorted (in nondecreasing order) sequence of column group indices has range 19 for $q \geq 23$. As a consequence, the template matrix in Fig. 3 in [8] can only prove an upper bound of 10 for $q \geq 73$.

Proposition 1: There is a codeword of weight 10 in $\mathcal{C}(q, 4, L, \boldsymbol{\zeta})$, $q \geq 13$, for all positive integers $L \geq 2$ and cutting vectors $\boldsymbol{\zeta}$ if one of the following conditions are true.

1) $\zeta_1 - \zeta_0 \geq 2$ and $\zeta_2 - \zeta_1 \geq 9$.
2) $\zeta_1 - \zeta_0 \geq 9$ and $\zeta_2 - \zeta_1 \geq 2$.
3) $\zeta_3 - \zeta_2 \geq 2$ and $q + \zeta_0 - \zeta_3 \geq 9$.
4) $\zeta_3 - \zeta_2 \geq 9$ and $q + \zeta_0 - \zeta_3 \geq 2$.
5) $\max(\zeta_1 - \zeta_0, \zeta_2 - \zeta_1, \zeta_3 - \zeta_2, q + \zeta_0 - \zeta_3) \geq 11$.

Proof: The proof is omitted due to lack of space. ∎

Lemma 3: The minimum distances $d_{\mathrm{opt}}(31, 4, L)$ and $d_{\mathrm{opt}}(37, 4, L)$, for $L \geq 2$, are both upper-bounded by 10.

Proof: For the case $q = 31$, the number of cutting vectors that satisfy none of the five conditions in Proposition 1 is only

35, while the total number of possible cutting vectors is $35\,960$. Thus, by running the algorithm from [15], adapted to the case of SC codes, on these 35 cases we have verified that there are indeed codewords of weight 10 for all cutting vectors, and the result follows.

For $q = 37$, there are no cutting vectors that satisfy none of the five conditions in Proposition 1, from which it follows that $d_{\text{opt}}(37, 4, L) \leq 10$ for all positive integers $L \geq 2$. ∎

By combining Lemmas 2 and 3, [2, Theorem 5], and [8, Corollary 4.2], we get the following theorem.

Theorem 2: For $L \geq 2$ and $q \geq 31$, the minimum distance $d_{\text{opt}}(q, 4, L) = 10$.

C. The Case $m = 5$

Theorem 3: The minimum distance $d_{\text{opt}}(q, 5, L)$, for $L \geq 2$ and $q \geq 59$, is upper-bounded by 12.

Proof: The proof follows the same main idea of the proofs of Lemmas 1 and 2 above using the template support matrix from Fig. 4 in [8], and is omitted for brevity. ∎

Note that there appears to be no equivalent to Proposition 1 (except for the last condition) for $m = 5$, since the template support matrix from Fig. 4 in [8] (which is used for the proof of Theorem 3) does not have the required structure. Also, the algorithm from [10] was not able to identify other nonequivalent template support matrices with a suitable structure. This has also been "confirmed" by the fact that for $q = 53$ we have been able to identify a cutting vector that gives a minimum distance of 14 (see Table III).

Theorem 4: The minimum distance $d_{\text{opt}}(q, 5, L)$, for $L \geq 2$ and $q \geq 29$, is upper-bounded by 16.

Proof: The proof follows the same main idea of the proofs of Lemmas 1 and 2 and Theorem 3 above using a template support matrix which was found using the algorithm from [10]. Due to lack of space, the actual template support matrix and the technical details of the proof are omitted. ∎

V. NUMERICAL RESULTS

In this section, we present some numerical results for the cases $m = 3$, 4, and 5.

A. The Case $m = 3$

By performing an exhaustive search over all cutting vectors $\boldsymbol{\zeta}$ the optimal minimum distance $d_{\text{opt}}(q, 3)$ was determined for $5 \leq q \leq 11$ (see Table III). For each of these values of q, the d_{min}-optimal cutting vector is not unique, i.e., there are several cutting vectors that give the best minimum distance. For instance, for $q = 5$, we found (for $L = 10$) the h_{min}-optimal (where h_{min} denotes stopping distance) cutting vectors displayed in Table I. For each cutting vector, in the second and fourth columns we have tabulated the corresponding minimum and stopping distance, respectively. The corresponding multiplicities are tabulated in the third (minimum distance) and fifth (stopping distance) columns. Note that the *optimal* cutting vector $\boldsymbol{\zeta} = (2, 3, 5)$ for $(3, 3)$ and $(4, 2)$ absorbing sets from [2] is also optimal when it comes to minimum/stopping distance, since it gives the optimal minimum/stopping distance

TABLE I
MINIMUM/STOPPING DISTANCE RESULTS FOR ARRAY-BASED SC-LDPC CODES FOR $q = 5$, $m = 3$, $L = 10$, AND DIFFERENT h_{min}-OPTIMAL CUTTING VECTORS

$\boldsymbol{\zeta}$	$d(5, 3, 10, \boldsymbol{\zeta})$	Mult.	$h(5, 3, 10, \boldsymbol{\zeta})$	Mult.
$(0, 1, 3)$	10	20	10	65
$(0, 2, 3)$	10	20	10	65
$(1, 2, 4)$	10	19	10	59
$(1, 3, 4)$	10	19	10	59
$(2, 3, 5)$	10	20	10	65
$(2, 4, 5)$	10	20	10	65

TABLE II
MINIMUM/STOPPING DISTANCE RESULTS FOR ARRAY-BASED SC-LDPC CODES FOR $q = 7$, $m = 4$, $L = 10$, AND DIFFERENT h_{min}-OPTIMAL CUTTING VECTORS

$\boldsymbol{\zeta}$	$d(7, 4, 10, \boldsymbol{\zeta})$	Mult.	$h(7, 4, 10, \boldsymbol{\zeta})$	Mult.
$(0, 2, 3, 5)$	14	30	14	401
$(0, 2, 4, 6)$	14	30	14	695
$(1, 3, 4, 6)$	14	29	14	393
$(1, 3, 5, 7)$	14	30	14	695
$(2, 4, 5, 7)$	14	30	14	401

of 10. However, it gives slightly more stopping sets of size 10 (see Table I). On the other hand, for $q = 7$, the *optimal* cutting vector $\boldsymbol{\zeta} = (2, 4, 6)$ for $(3, 3)$ and $(4, 2)$ absorbing sets from [2] is *not* optimal when it comes to minimum distance, since it gives a minimum distance of only 6 (the optimal minimum distance is 8). For $q = 5, 7$, d_{min}-optimal cutting vectors that also give the lowest possible minimum distance multiplicity, denoted as $(d_{\text{min}}, N_{d_{\text{min}}})$-optimal cutting vectors, are displayed within the parentheses in the sixth column of Table III (first and second row, respectively).

For $q \geq 13$, it follows from Theorem 1 that the optimal minimum distance is 6. Also, all cutting vectors give a minimum distance of 6, but possibly with different multiplicities.

B. The Case $m = 4$

By performing an exhaustive search over all cutting vectors $\boldsymbol{\zeta}$ (with the help of Proposition 1 to reduce the search space when $q > 11$) the optimal minimum distance $d_{\text{opt}}(q, 4)$ was determined for $5 \leq q \leq 29$. For each of these values of q, the d_{min}-optimal cutting vector is not unique, i.e., there are several cutting vectors that give the best minimum distance. For instance, for $q = 7$, we found (for $L = 10$) the h_{min}-optimal cutting vectors displayed in Table II. For each cutting vector, in the second and fourth columns we have tabulated the corresponding minimum and stopping distance, respectively. The corresponding multiplicities are tabulated in the third (minimum distance) and fifth (stopping distance) columns. For $5 \leq q \leq 29$, the optimal minimum distance and corresponding $(d_{\text{min}}, N_{d_{\text{min}}})$-optimal cutting vectors (displayed within the parentheses) are given in the fourth column of Table III (the first eight rows, respectively). For $q \geq 31$, it follows from Theorem 2 that $d_{\text{opt}}(q, 4) = 10$.

C. The Case $m = 5$

For $q = 7$, we have performed an exhaustive search over all possible cutting vectors. The optimal minimum distance $d_{\text{opt}}(7, 5)$ is 14 and a $(d_{\text{min}}, N_{d_{\text{min}}})$-optimal cutting vector is displayed within the parentheses in the second column of

TABLE III
Optimal Minimum Distance Results for Array-Based SC-LDPC Codes for Different Values of q and m. Values in Bold Are New Results, While Non-Bold Values Are Taken From the Literature. The Results Are Compared With Those of Array-Based Uncoupled LDPC Codes Taken From [7, 8, 10]. Within the Parentheses the Corresponding Cutting Vectors Are Displayed. Cutting Vectors in Italics Are $d_{\min}$-Optimal, While Cutting Vectors in Bold Are $(d_{\min}, N_{d_{\min}})$-Optimal

q	$d_{\mathrm{opt}}(q,5)$	$d(q,5)$	$d_{\mathrm{opt}}(q,4)$	$d(q,4)$	$d_{\mathrm{opt}}(q,3)$	$d(q,3)$
5			**10** (1, 2, 3, 4)	8	**10** (1, 2, 4)	6
7	**14** (**1**, **2**, **3**, **4**, **6**)	12	**14** (1, 3, 4, 6)	8	**8** (**1**, **2**, **5**)	6
11	**22** (0, 1, 2, 5, 8)	10	**14** (1, 3, 6, 8)	10	**6**	6
13	**24** (0, 2, 5, 7, 10)	12	**12** (0, 3, 7, 9)	10	**6**	6
17	**20**, **22**, or **24** (0, 4, 7, 11, 14)	12	**12** (2, 5, 10, 13)	10	**6**	6
19	**20** or **22** (0, 4, 8, 12, 16)	12	**12** (3, 7, 12, 16)	10	**6**	6
23	**20** (5, 10, 15, 19, 23)	12	**12** (4, 8, 15, 19)	10	**6**	6
29	**16** (8, 11, 18, 26, 29)	12	**12** (**4**, **11**, **18**, **25**)	10	**6**	6
31	**16** (9, 15, 21, 24, 31)	12	**10**	10	**6**	6
37	**16** (0, 10, 19, 28, 30)	12	**10**	10	**6**	6
41	**16** (0, 9, 18, 25, 32)	12	**10**	10	**6**	6
43	**16** (0, 8, 19, 24, 33)	12	**10**	10	**6**	6
47	**16** (7, 16, 26, 36, 47)	12	**10**	10	**6**	6
53	**14** (**0**, **11**, **22**, **33**, **42**)	12	**10**	10	**6**	6
$59-79$	**12**	12	**10**	10	**6**	6
>79	**10** or **12**	10 or 12	**10**	10	**6**	6

Table III (second row). For $q = 11$ and 13, the optimal minimum distance is as high as 22 and 24, respectively, and $d_{\min}$-optimal cutting vectors are shown within the parentheses in the second column of Table III (third and fourth row, respectively). For $q = 17$ and 19, upper and lower bounds on the optimal minimum distance, tabulated in the second column of Table III, have been established from an exhaustive search over all cutting vectors. The displayed values are bounds since we were not able to exhaustively enumerate all codewords of weight at most τ, when $\tau \geq 20$, for a given cutting vector, for these two values of q. Within the parentheses a corresponding cutting vector (which establishes the upper bound) is also tabulated. The lower bounds were determined by running the algorithm from [15], adapted to the case of SC codes, for the specific cutting vectors displayed within the parentheses. For $23 \leq q \leq 53$, the tabulated values (in the second column of Table III) are again the exact values of $d_{\mathrm{opt}}(q,5)$ and also (within the parentheses) $d_{\min}$-optimal ($(d_{\min}, N_{d_{\min}})$-optimal for $q = 53$) cutting vectors are displayed. For $59 \leq q \leq 79$, $d_{\mathrm{opt}}(q,5) = 12$, which follows from Theorem 3, [10, Table I], and Theorem 5 in [2]. For $q > 79$, it follows from Theorem 3, Corollary 4.4 in [8], and Theorem 5 in [2] that $d_{\mathrm{opt}}(q,5)$ is either 10 or 12, although we conjecture it to be 12.

VI. Conclusion and Future Work

In this work, we have studied in detail the minimum distance of array-based SC-LDPC codes. Several tight upper bounds on the optimal minimum distance for coupling length $L \geq 2$ and $m = 3, 4, 5$, that are independent of q and that are valid for all values of $q \geq q_0$ where q_0 depends on m, have been presented. Furthermore, we have conducted an exhaustive search over all cutting vectors for small values of q ($m = 3, 4, 5$) which shows that by carefully selecting the cutting vector, the minimum distance (when q is not very large) can be significantly increased, especially for $m = 5$.

An interesting topic for future work is to consider the correlation with absorbing sets. In particular, will a $(d_{\min}, N_{d_{\min}})$-optimal cutting vector also be *close-to-optimal* when it comes to problematic absorbing sets, and/or vice-versa?

References

[1] M. Baldi, G. Cancellieri, and F. Chiaraluce, "Array convolutional low-density parity-check codes," *IEEE Commun. Lett.*, vol. 18, no. 2, pp. 336–339, Feb. 2014.

[2] D. G. M. Mitchell, L. Dolecek, and D. J. Costello, Jr., "Absorbing set characterization of array-based spatially coupled LDPC codes," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Honolulu, HI, Jun./Jul. 2014, pp. 886–890.

[3] J. L. Fan, "Array codes as low-density parity-check codes," in *Proc. 2nd Int. Symp. Turbo Codes & Rel. Topics*, Brest, France, Sep. 2000, pp. 543–546.

[4] 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.

[5] S. Kudekar, T. Richardson, and R. L. Urbanke, "Spatially coupled ensembles universally achieve capacity under belief propagation," *IEEE Trans. Inf. Theory*, vol. 59, no. 12, pp. 7761–7813, Dec. 2013.

[6] T. Mittelholzer, "Efficient encoding and minimum distance bounds of Reed-Solomon-type array codes," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Lausanne, Switzerland, Jun./Jul. 2002, p. 282.

[7] K. Yang and T. Helleseth, "On the minimum distance of array codes as LDPC codes," *IEEE Trans. Inf. Theory*, vol. 49, no. 12, pp. 3268–3271, Dec. 2003.

[8] K. Sugiyama and Y. Kaji, "On the minimum weight of simple full-length array LDPC codes," *IEICE Trans. Fundamentals*, vol. E91-A, no. 6, pp. 1502–1508, Jun. 2008.

[9] M. Esmaeili and M. J. Amoshahy, "On the stopping distance of array code parity-check matrices," *IEEE Trans. Inf. Theory*, vol. 55, no. 8, pp. 3488–3493, Aug. 2009.

[10] E. Rosnes, M. A. Ambroze, and M. Tomlinson, "On the minimum/stopping distance of array low-density parity-check codes," *IEEE Trans. Inf. Theory*, vol. 60, no. 9, pp. 5204–5214, Sep. 2014.

[11] L. Dolecek, Z. Zhang, V. Anantharam, M. J. Wainwright, and B. Nikolic, "Analysis of absorbing sets and fully absorbing sets of array-based LDPC codes," *IEEE Trans. Inf. Theory*, vol. 56, no. 1, pp. 181–201, Jan. 2010.

[12] S. Ölçer, "Decoder architecture for array-code-based LDPC codes," in *Proc. IEEE Global Telecommun. Conf. (GLOBECOM)*, vol. 4, San Francisco, CA, Dec. 2003, pp. 2046–2050.

[13] P. Bhagawat, M. Uppal, and G. Choi, "FPGA based implementation of decoder for array low-density parity-check codes," in *Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Processing (ICASSP)*, vol. 5, Philadelphia, PA, Mar. 2005, pp. 29–32.

[14] A. E. Pusane, R. Smarandache, P. O. Vontobel, and D. J. Costello, Jr., "Deriving good LDPC convolutional codes from LDPC block codes," *IEEE Trans. Inf. Theory*, vol. 57, no. 2, pp. 835–857, Feb. 2011.

[15] E. Rosnes, Ø. Ytrehus, M. A. Ambroze, and M. Tomlinson, "Addendum to "An efficient algorithm to find all small-size stopping sets of low-density parity-check matrices"," *IEEE Trans. Inf. Theory*, vol. 58, no. 1, pp. 164–171, Jan. 2012.