

Design of LDPC Codes for the q -ary Partial Erasure Channel

Rami Cohen and Yuval Cassuto

Department of Electrical Engineering
Technion - Israel Institute of Technology
Technion City, Haifa 3200003, Israel

Email: rc@tx.technion.ac.il, ycassuto@ee.technion.ac.il

Abstract—In this paper, we discuss practical design of low-density parity-check (LDPC) codes for the q -ary partial erasure channel (QPEC). This channel is an extension of the binary erasure channel (BEC), where partial information on the output is available. We provide a linear programming (LP) optimization for the design of good degree distributions, and compare our code design results to codes obtained using an LP optimization formulated for the BEC. We show superior performance in terms of code rate and complexity, when designing an LDPC code for a desired decoding threshold.

I. INTRODUCTION

Linear codes constructed from low-density matrices, known as low-density parity-check (LDPC) codes [1], [2], offer low complexity of implementation, with good performance under iterative decoding. These codes were shown to achieve performance close to the capacity for several important channels, using efficient decoding algorithms [3], [4]. Recently, the use of q -ary LDPC codes was suggested for the q -ary partial erasure channel (QPEC) [5]. In this channel, the transmitted symbol is either completely known with probability $1 - \varepsilon$ or known up to $M \leq q$ values with probability ε . The latter case is termed as a *partial erasure* event, since there is only a partial uncertainty in the output. The binary erasure channel (BEC) and the q -ary erasure channel (QEC) can be thought as special cases of the QPEC when $M = q$.

In [5], the density-evolution method was used to provide decoding performance evaluation of q -ary LDPC codes over the QPEC under iterative decoding. The decoding threshold, defined as the maximal partial-erasure probability such that the asymptotic probability of decoding failure is zero, was then derived using the density-evolution equations. Compared to the BEC and the binary messages (erasure/non-erasure) passed on the iterative decoding process, there is a spectrum of possible messages in the QPEC, due to possible partial erasures. This results in q -dimensional density-evolution equations, whose exact analysis is difficult. Therefore, the design of good LDPC codes (e.g., in terms of rate versus decoding threshold) for the QPEC a difficult task.

In this work, we formulate a linear programming (LP) optimization for designing good LDPC code degree distributions for the QPEC. This is based on a single-letter expression derived in [6] for finding an upper bound on the QPEC decoding threshold. We suggest an optimization process for finding degree distributions with a desired decoding threshold. We compare our optimization method to a BEC-based optimization process, and show that it yields better codes than

those designed with the known linear-programming degree-distribution optimization for the BEC.

This paper is structured as follows. We review the QPEC channel model and its density evolution equations in Section II. Code design tools are discussed in Section III. Finally, conclusions are given in Section IV.

II. PRELIMINARIES

A. Channel model

The q -ary partial erasure channel (QPEC), introduced in [5], is defined as follows. Let X be the transmitted symbol taken from the alphabet $\mathcal{X} = \{0, 1, \dots, q - 1\}$. We assume that q is a prime or a prime power, such that the symbols in \mathcal{X} can be considered as the elements of the finite field $\text{GF}(q)$. For each $x \in \mathcal{X}$, define the set $\left\{ ?_x^{(i)} \right\}_{i=1}^{i_{\max}}$, $i_{\max} = \binom{q-1}{M-1}$, where each *super-symbol* $?_x^{(i)}$ is a set of size M that contains the symbol x and $M - 1$ other symbols, taken from $\mathcal{X} \setminus \{x\}$. Let Y be the received symbol with the output alphabet $\mathcal{Y} = \left\{ \mathcal{X} \bigcup_{x=0}^{q-1} \left\{ ?_x^{(i)} \right\}_{i=1}^{i_{\max}} \right\}$. The transition probabilities governing the QPEC are as follows:

$$\Pr(Y = y | X = x) = \begin{cases} 1 - \varepsilon, & y = x \\ \varepsilon / i_{\max}, & y = ?_x^{(i)} \end{cases} \quad (1)$$

for $i = 1, 2, \dots, i_{\max}$, where $0 \leq \varepsilon \leq 1$ is the (partial) erasure probability.

That is, the output of the channel can be either the transmitted symbol with probability $1 - \varepsilon$ (corresponding to a non-erasure event), or a set of M symbols containing the transmitted symbol with probability ε (corresponding to a partial erasure event). In the latter case, it can be any of i_{\max} possible sets (of size M), each having the same probability ε / i_{\max} . As an example, assume that $q = 4$, $M = 3$, and the symbol 0 was transmitted. Then 0 is received with probability $1 - \varepsilon$, where each of the sets $?_0^{(1)} = \{0, 1, 2\}$, $?_0^{(2)} = \{0, 1, 3\}$ and $?_0^{(3)} = \{0, 2, 3\}$, is received with probability $1 - \varepsilon$.

B. Density-evolution equations

A $\text{GF}(q)$ LDPC code is defined in a similar way to its binary counterpart, by a sparse parity-check matrix, or equivalently by a Tanner graph [7]. This graph is bipartite, with variable (left) nodes corresponding to symbols of the codeword, and check (right) nodes corresponding to parity-check equations. The codeword symbols and the labels on the graph edges are

taken from $\text{GF}(q)$. In addition, the parity-check equations are defined using $\text{GF}(q)$ arithmetic.

A message-passing decoder for $\text{GF}(q)$ LDPC codes over the QPEC was proposed in [5], generalizing the BEC/QEC iterative decoder [3]. There are two types of messages: *check to variable* (CTV) messages and *variable to check* (VTC) messages. At iteration $l = 0$, channel information is sent from variable to check nodes. In the subsequent iterations, a CTV message from a check node contains all possible assignments of the target variable node given the contents of the incoming VTC messages, such that the check-node associated parity-check equation is satisfied. A VTC message from a variable node is simply the intersection of the channel information and the incoming CTV messages to the variable node.

The density evolution method [8], [9] allows to track the asymptotic (in terms of codeword length) probability of decoding failure as a function of decoding iteration. The QPEC density-evolution equations governing the probability of decoding failure were derived in [5], and are provided here for completeness. Define the probability vector $w^{(l)}$, where $w_m^{(l)} (m = 1, 2, \dots, q)$ denotes the probability that a CTV message at iteration l is of size m . The probability vector $z^{(l)}$ is defined for VTC messages in a similar manner. In addition, define the degree-distribution polynomials $\lambda(x) = \sum_i \lambda_i x^{i-1}$

and $\rho(x) = \sum_i \rho_i x^{i-1}$, where $\lambda_i (\rho_i)$ denotes the fraction of edges connected to a variable (check) node of degree i [3], and $d_v (d_c)$ is the maximal variable (check) node degree. The QPEC density-evolution equations are:

$$w_m^{(l)} = \sum_{i=1}^{d_c} \rho_i \cdot \sum_{\substack{\{m_j\}_{j=1}^{i-1}: \\ m_j \leq M}} \left(\prod_{j=1}^{i-1} z_{m_j}^{(l-1)} \right) \cdot P_m \left(\{m_j\}_{j=1}^{i-1} \right), \quad (2)$$

$$\begin{aligned} z_m^{(l)} &= \delta[m-1] \cdot (1-\varepsilon) \\ &+ \varepsilon \sum_{i=1}^{d_v} \lambda_i \cdot \sum_{\substack{\{m_j\}_{j=1}^{i-1}: \\ m_j \leq q}} \left(\prod_{j=1}^{i-1} w_{m_j}^{(l)} \right) \cdot Q_m \left(\{m_j\}_{j=1}^{i-1}, M \right), \end{aligned} \quad (3)$$

where $P_m \left(\{m_j\}_{j=1}^{i-1} \right)$ denotes the probability for a CTV message of size m , given incoming VTC messages of sizes $\{m_j\}_{j=1}^{i-1}$, and $Q_m \left(\{m_j\}_{j=1}^{i-1}, M \right)$ is defined similarly for VTC messages (M denotes the channel information set size). $\delta[m]$ denotes the discrete Dirac delta function.

The density-evolution equations (2)-(3) are multi-dimensional. Moreover, the calculation of P_m of (2) is computationally difficult unless approximations are used. A single-letter expression for the evolution of $z_M^{(l)}$ was suggested in [6], leading to an upper bound on the QPEC decoding threshold assuming that $M > q/2$. Recall that $z_M^{(l)}$ denotes the probability for a VTC message of size M at iteration l , where M is the maximal size possible for a VTC message, representing no-progress from the start of decoding. Denote by $\rho'(x)$ the derivative of $\rho(x)$ with respect to x . The upper bound on the QPEC decoding threshold, denoted ε^* , is provided in the following theorem.

Theorem 1. [6] For a fixed ε , define the function $h_\varepsilon(x) = \varepsilon \lambda(1 - \rho(1-x) - x\rho'(1-x))$, and consider the i^{th} composition of $h_\varepsilon(x)$ with itself, denoted $h_\varepsilon^i(x)$. Then, the QPEC decoding threshold for $M > q/2$ is upper bounded by

$$\varepsilon^* = \sup \left\{ \varepsilon \in [0, 1] : \lim_{l \rightarrow \infty} h_\varepsilon^l(\varepsilon) = 0 \right\}. \quad (4)$$

In the following, we obtain an equivalent definition of ε^* , and later use it to derive a linear programming optimization for the design of good degree distributions. We later refine the proposed optimization, by enforcing additional constraints on the check-node degree.

III. CODE DESIGN USING LINEAR PROGRAMMING

In this section we discuss code-design tools for the QPEC. The main tool is a linear programming (LP) optimization derived from the threshold upper bound ε^* of Equation (4). For the formulation of this LP optimization we first derive an equivalent definition for the ε^* . This definition is obtained by extending the fixed-point characterization of the BEC threshold [10], as follows.

Theorem 2. For the QPEC with $M > q/2$,

$$\varepsilon^* = \sup \{ \varepsilon \in [0, 1] : x = h_\varepsilon(x) \text{ has no solution } x \text{ in } (0, 1) \}. \quad (5)$$

Proof. Consider the recursive equation $x_l(x_0) = h_\varepsilon(x_{l-1}(x_0))$ for $\varepsilon, x_0 \in [0, 1]$ and for $l \geq 1$. We will prove that $x_l(x_0)$ converges to a solution of $x = h_\varepsilon(x)$. First, if $x_0 = 0$ or $\varepsilon = 0$ then x_l converges to 0, which is a fixed point of $h_\varepsilon(x)$. For other values of x_0 and ε , note that $h_\varepsilon(x)$ is an increasing function function of x [6]. Therefore, assuming that $x_l \geq x_{l-1}$, the following inequality holds:

$$x_{l+1} = h_\varepsilon(x_l) \geq h_\varepsilon(x_{l-1}) = x_l. \quad (6)$$

If $x_l \leq x_{l-1}$, The inequality sign of (6) flips. In either case, we conclude by induction that x_l is a monotone sequence, and since $h_\varepsilon(x)$ is bounded by ε we deduce that x_l converges in the interval $[0, \varepsilon]$.

By the continuity of $h_\varepsilon(x)$, we have that

$$\lim_{l \rightarrow \infty} x_l = \lim_{l \rightarrow \infty} h_\varepsilon(x_{l-1}), \quad (7)$$

meaning that x_l converges to a solution of the equation $x = h_\varepsilon(x)$. Thinking of x_0 as $z_M^{(0)}$, which equals ε , we can write:

$$\lim_{l \rightarrow \infty} h_\varepsilon(z_M^{(l-1)}) = \lim_{l \rightarrow \infty} h_\varepsilon^l(\varepsilon), \quad (8)$$

where $h_\varepsilon^l(\varepsilon)$ denotes the l -th composition of $h_\varepsilon(x)$ with itself (see Theorem 1) at $x = \varepsilon$. If there is a solution of the equation $x = h_\varepsilon(x)$ that is larger than 0, then ε is necessarily larger than ε^* , according to Equation (4). Thus, for $\varepsilon \leq \varepsilon^*$, the only fixed point of $h_\varepsilon(x)$ is at 0, requiring equivalently that $x = h_\varepsilon(x)$ has no solution in the interval $(0, 1]$. \square

An important outcome from Theorem 2 is that ε^* can be found by drawing $h_\varepsilon(x) - x$ as a function of x for $x \in (0, 1]$, for each ε , where ε^* is found as the largest ε such that $h_\varepsilon(x) - x \leq 0$. This is demonstrated in Figure 1, for a regular $(3, 6)$ LDPC code. Based on Theorem 2, we formulate an LP optimization for determining good (in terms of code rate)

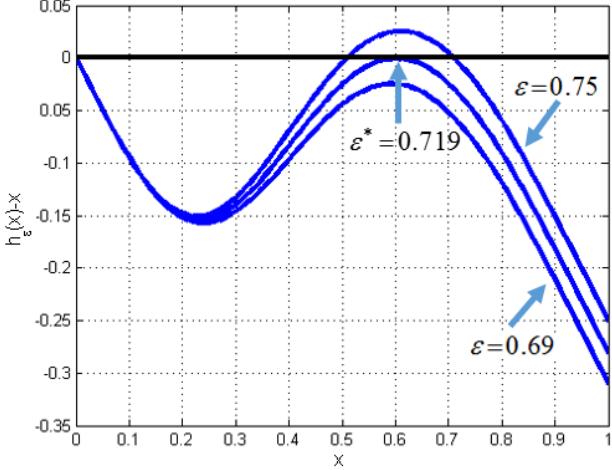


Fig. 1: Graphical determination of ε^* for a (3, 6) LDPC code. The graph of $h_\varepsilon(x) - x$ tangentially touches the horizontal axis for $\varepsilon^* = 0.719$.

variable-node degree distribution $\lambda(x)$ for a given $\rho(x)$, ε^* and maximum variable-node degree d_v , assuming that $M > q/2$. Note that because decoding performance improves as M decreases, degree distributions providing a certain threshold for $M > q/2$ will provide at least the same threshold for $M \leq q/2$.

For the formulation of the LP optimization, note that the design rate of a code ensemble with degree-distribution pair $\lambda(x)$ and $\rho(x)$ is [3]:

$$r = 1 - \left(\sum_i \frac{\rho_i}{i} \right) / \left(\sum_i \frac{\lambda_i}{i} \right). \quad (9)$$

We set a maximum constraint d_v on variable-node degrees, as usual [3], to control implementation complexity and convergence speed. According to the graphical determination of ε^* described earlier, the condition for degree distributions whose threshold is upper bounded by ε^* is that $h_{\varepsilon^*}(x) - x \leq 0$ for $x \in (0, 1]$. Therefore, we formulate the following LP optimization for the QPEC:

$$\max_{\lambda} \left\{ \sum_{i=2}^{d_v} \frac{\lambda_i}{i} : \lambda_i \geq 0, \sum_{i=2}^{d_v} \lambda_i = 1, h_{\varepsilon^*}(x) - x \leq 0; x \in (0, 1] \right\} \quad (10)$$

which maximizes the rate for a desired ε^* . Note the similarity between (10) and the known degree-distribution LP optimization for the BEC threshold [3]. The key difference is in using in (10) the function $h_{\varepsilon^*}(x)$ specially developed for the QPEC, instead of the function $f_\varepsilon(x) = \varepsilon \cdot \lambda(1 - \rho(1 - x))$ derived from the BEC density-evolution equation.

The LP optimization in (10) can also be formulated from check-node perspective. For this formulation, we rewrite the condition $h_{\varepsilon^*}(x) - x \leq 0$, as follows. Define $y = \lambda^{-1}(\frac{x}{\varepsilon})$ (note that the inverse of $\lambda(x)$ exists since $\lambda(x)$ is monotonically increasing). Apply $\lambda^{-1}(\cdot)$ to both sides of the inequality $h_{\varepsilon^*}(x) - x \leq 0$. Then, the condition for degree-distribution pair whose decoding threshold is at most ε^* is

$$1 - \rho(1 - \varepsilon^* \lambda(y)) - \varepsilon^* \lambda(y) \rho'(1 - \varepsilon^* \lambda(y)) \leq y, \quad (11)$$

for $y \in (0, 1]$. Denoting the left-hand side of the inequality in (11) by $\hat{h}_{\varepsilon^*}(y)$, we get the following LP optimization:

$$\min_{\rho} \left\{ \sum_{i=2}^{d_c} \frac{\rho_i}{i} : \rho_i \geq 0, \sum_{i=2}^{d_c} \rho_i = 1, \hat{h}_{\varepsilon^*}(y) - y \leq 0; y \in (0, 1] \right\}. \quad (12)$$

Here as well the LP optimization of (12) is similar to the known BEC optimization, with $\hat{f}_\varepsilon(y) = 1 - \rho(1 - \varepsilon \cdot \lambda(y))$ replaced by the QPEC-special function $\hat{h}_{\varepsilon^*}(y)$.

As noted earlier, the QPEC LP optimization provides a degree-distribution pair with a decoding threshold upper bounded by ε^* . However, we are interested in degree distributions with an exact threshold $\varepsilon^{\text{QPEC}}$. One strategy for obtaining such degree distributions is as follows. Choose ε^* that is larger than the desired QPEC threshold $\varepsilon^{\text{QPEC}}$, and solve the QPEC LP optimization. Check the actual QPEC decoding threshold of the optimized degree distributions using the density evolution equations (2)-(3). If this threshold is smaller than $\varepsilon^{\text{QPEC}}$, increase ε^* and repeat the process. Otherwise, decrease ε^* and repeat the process.

In a similar way, a QPEC decoding threshold $\varepsilon^{\text{QPEC}}$ can be sought by using the well-known BEC LP optimization. Because the BEC is a degraded version of the QPEC, here we will choose a target threshold ε^{BEC} smaller than the desired $\varepsilon^{\text{QPEC}}$. We then similarly calculate the true QPEC threshold of the resulting degree distributions using (2)-(3). If this threshold is larger than $\varepsilon^{\text{QPEC}}$, we decrease ε^{BEC} and repeat the process. Otherwise, we increase ε^{BEC} and repeat the process.

It turns out that using the first approach – optimizing for ε^* – can result in better codes compared to the second approach using the BEC optimization. In the sequel we show this by numerical examples. The intuition behind this improvement is that the new LP optimization for ε^* better captures the decoding performance for QPECs with $M < q$. Before discussing the efficiency of the LP optimization tools for QPEC code design, we want to better understand the relations between the target thresholds input to the LP optimizations and the actual QPEC threshold achieved by the resulting degree distributions.

To this end we have two LP optimizations; one of them (QPEC ε^*) providing a degree distribution with QPEC threshold at most its target, and another (BEC) providing a degree distribution with QPEC threshold at least its target. So each run of the QPEC ε^* optimizer provides an upper bound on the achievable rate for codes with the target QPEC threshold, and each run of the BEC optimizer provides a lower bound on the achievable rate for codes with the target QPEC threshold. An example for that is given in Figure 2 for $\rho(x) = x^5$ and $d_v = 5$. The true optimal rate over the QPEC lies somewhere in between the two curves of Figure 2.

We now show the benefit of the new QPEC LP optimization in achieving better codes than those obtained using the BEC LP optimization. As an example, assume that $\rho(x) = x^5$, $d_v = 5$ and the desired $\varepsilon^{\text{QPEC}}$ is 0.6. The optimization results are provided in Table I when the QPEC LP optimization is used, and in Table II when the BEC LP optimization is used. We concentrate here on QPECs with $M = \lfloor q/2 \rfloor + 1$ for several values of q (this value of M is the smallest satisfying $M >$

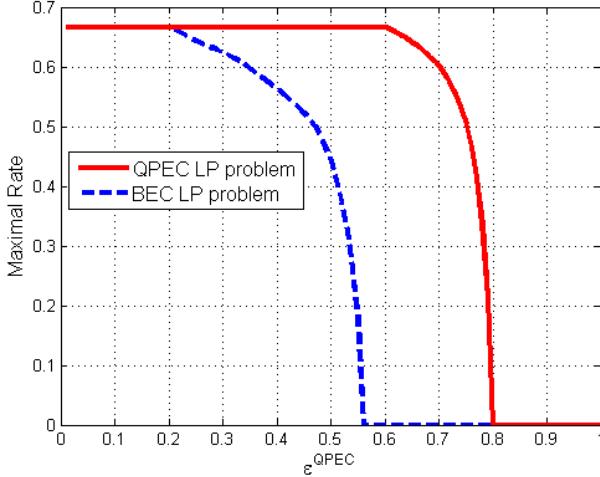


Fig. 2: Lower and upper bounds on the achievable code rate as a function of QPEC decoding threshold, for $\rho(x) = x^5$ and $d_v = 5$.

$q/2$). In the optimization process, we searched using iterative instantiations of the optimizers, for values of ε^* and ε^{BEC} such that the degree distributions obtained in the corresponding LP optimizations provide a QPEC decoding threshold $\varepsilon^{\text{QPEC}}$.

When comparing the results in Table I and Table II, we observe that for all the parameters checked the rates achieved by the QPEC optimizer are strictly better than the rates resulting from the BEC optimizer. For some parameters the difference is quite significant. Another interesting observation is that in some cases the BEC optimizer required a $\lambda(x)$ polynomial with more non-zero coefficients than the QPEC optimizer. This implies that the QPEC optimizer provides degree distributions that are easier to implement in practice.

TABLE I: QPEC optimization results for $\rho(x) = x^5$, $d_v = 5$ and a desired QPEC decoding threshold $\varepsilon^{\text{QPEC}} = 0.6$.

q	M	ρ_i, λ_i	ε^*	Rate
3	2	$\rho_6 = 1$ $\lambda_2 = 0.644$ $\lambda_5 = 0.356$	0.718	0.576
4	3	$\rho_6 = 1$ $\lambda_2 = 0.193$ $\lambda_5 = 0.807$	0.778	0.354
5	3	$\rho_6 = 1$ $\lambda_2 = 0.489$ $\lambda_5 = 0.511$	0.751	0.519
7	4	$\rho_6 = 1$ $\lambda_2 = 0.372$ $\lambda_5 = 0.628$	0.763	0.465

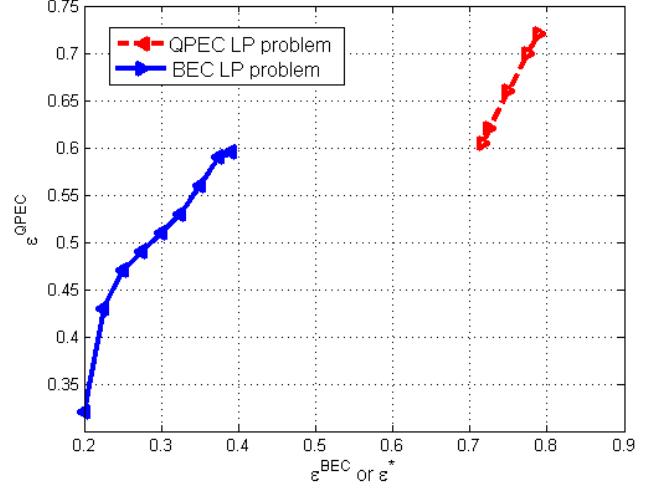


Fig. 3: Optimization process, for $\rho(x) = x^5$ and $d_v = 5$ and a desired QPEC ($q = 3, M = 2$) decoding threshold $\varepsilon^{\text{QPEC}} = 0.6$.

TABLE II: BEC optimization results for $\rho(x) = x^5$, $d_v = 5$ and a desired QPEC decoding threshold $\varepsilon^{\text{QPEC}} = 0.6$.

q	M	ρ_i, λ_i	ε^{BEC}	Rate
3	2	$\rho_6 = 1$ $\lambda_2 = 0.517$ $\lambda_3 = 0.099$ $\lambda_4 = 0.384$	0.391	0.569
4	3	$\rho_6 = 1$ $\lambda_2 = 0.157$ $\lambda_5 = 0.843$	0.532	0.325
5	3	$\rho_6 = 1$ $\lambda_2 = 0.437$ $\lambda_3 = 0.056$ $\lambda_5 = 0.507$	0.464	0.508
7	4	$\rho_6 = 1$ $\lambda_2 = 0.345$ $\lambda_5 = 0.655$	0.492	0.451

An illustration of the iterative optimization process is provided in Figure 3. The plot shows the sequence of optimization runs of the QPEC optimizer (right), and the sequence of runs for the BEC optimizer (left). The former approaches the target of $\varepsilon^{\text{QPEC}} = 0.6$ from above, and the latter from below. Reaching the desired QPEC threshold took fewer optimization instances with the QPEC optimizer than with the BEC optimizer.

A. Refined LP considering the specific value of M

A disadvantage of the QPEC LP optimization presented above is its independence of the exact values of q and M . That is, assuming a fixed q , the QPEC LP optimizes the degree distribution for a general $M > q/2$, without taking into account the specific M we need to design for. Therefore, in this sub-section we add to the QPEC LP an additional constraint,

which depends explicitly on q and M , and (loosely speaking) can help guiding the optimizer to degree distributions that are more beneficial to the specific M in question. Assuming a given M satisfying $M > q/2$, there is a certain fraction of incoming VTC message sizes that satisfy the q -condition [5]. This condition is satisfied if there is at least one pair of incoming VTC messages whose sum of sizes exceeds q . In this case, the outgoing CTV message is of size q , meaning no decoding progress. Intuitively, if this fraction is high (say close to 1), decoding is not expected to make progress. The following theorem calculates this fraction exactly as a function of M , q , and the check-node degree i .

Theorem 3. Define the vector $\tau = (\tau_1, \tau_2, \dots, \tau_{i-1})$ where τ_j ($j = 1, 2, \dots, i-1$) is the size of the j^{th} incoming message to a check node of degree i . Then, the fraction of such vectors whose elements satisfy the q -condition is:

$$\xi(i, M, q) = \frac{M^{i-1} - \lfloor \frac{q}{2} \rfloor^{i-1} - (i-1) \cdot \sum_{\tau^{\max}} (q - \tau^{\max})^{i-2}}{M^{i-1}}, \quad (13)$$

where the summation is over $\lfloor q/2 \rfloor + 1 \leq \tau^{\max} \leq M$.

Proof. There are M^{i-1} possible vectors that can describe incoming message sizes to a check node of degree i , since a VTC message size is upper bounded by M . The q -condition is satisfied if there are two indices j_1, j_2 ($j_1 \neq j_2$) such that $\tau_{j_1} + \tau_{j_2} > q$. This condition can not be satisfied if the incoming message sizes are upper bounded by $\lfloor q/2 \rfloor$, since no sum of two elements can exceed q in this case. The number of such vectors is $\lfloor q/2 \rfloor^{i-1}$. Similarly, vectors whose single maximum element τ^{\max} satisfies $\lfloor q/2 \rfloor + 1 \leq \tau^{\max} \leq M$ and remaining elements are at most $q - \tau^{\max}$ represent message sizes that do not satisfy the q -condition. There are $(i-1) \cdot (q - \tau^{\max})^{i-2}$ such vectors, since the maximum can be placed in one of $i-1$ positions. Finally, Equation (13) is obtained by subtracting the number of vectors that do not satisfy the q -condition from the number of possible vectors, and normalizing by the number of possible vectors. \square

As an example, $\xi(i, M, q)$ for $q = 8$ is provided in Figure 4. For a check node with degree i , an initial fraction of ε incoming messages is erased. Therefore, assuming a desired decoding threshold $\varepsilon^{\text{QPEC}}$, at most $\lceil \varepsilon^{\text{QPEC}} \cdot i \rceil$ incoming messages may satisfy the q -condition. This leads us to formulate an extended QPEC LP optimization, with an additional constraint that the fraction of incoming VTC message sizes satisfying the q -condition is bounded from above by a pre-defined value $\hat{\delta}$ ($\hat{\delta} \in (0, 1]$):

$$\min_{\rho} \left\{ \begin{array}{l} \sum_{i=2}^{d_c} \frac{\rho_i}{i} : \rho_i \geq 0, \sum_{i=2}^{d_c} \rho_i = 1, \hat{h}_{\varepsilon^*}(y) - y \leq 0; y \in (0, 1], \\ \sum_{i=2}^{d_c} \rho_i \cdot \xi(\lceil \varepsilon^{\text{QPEC}} \cdot i \rceil, M, q) - \hat{\delta} \leq 0. \end{array} \right. \quad (14)$$

The role of $\hat{\delta}$ can be thought as an additional control over the threshold of the optimized degree-distribution pair. As $\hat{\delta}$ becomes smaller, check-node outgoing messages of size q are less likely to occur, and the exact threshold increases. The LP optimization in (14) can be used for code design, similarly to the optimization process described earlier. One may choose

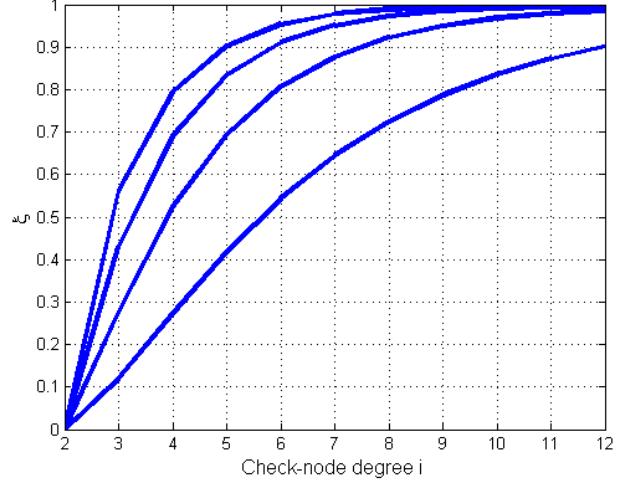


Fig. 4: ξ as a function of check-node degree i , for $q = 8$ and $M = 5, 6, 7, 8$ (from bottom up).

ε^* and change $\hat{\delta}$ to obtain a desired QPEC threshold, or vice versa. An important advantage of (14) is its explicit dependency on q and M , through ξ .

IV. CONCLUSIONS

In this work, we suggested LP optimizations for finding LDPC codes with a desired QPEC decoding threshold. We compared our results to degree distributions obtained using a known BEC LP optimization, and showed better performance in terms of code rate and decoding complexity. This work provides a first step in the design of good LDPC codes for the QPEC, which is a part of our ongoing research.

V. ACKNOWLEDGEMENT

This work was partly done under a joint ISF-UGC project. In addition, it was supported by a Marie Curie CIG grant, by the Israel Ministry of Science and Technology, and by a GIF Young Investigator grant.

REFERENCES

- [1] R. G. Gallager, "Low-density parity-check codes," *IRE Transactions on Information Theory*, vol. 8, no. 1, pp. 21–28, 1962.
- [2] ———, *Low-Density Parity Check Codes*. MIT Press, 1963.
- [3] T. Richardson and R. Urbanke, *Modern Coding Theory*. Cambridge University Press, 2008.
- [4] S.-Y. Chung, G. D. Forney, Jr., T. J. Richardson, and R. Urbanke, "On the design of low-density parity-check codes within 0.0045 dB of the shannon limit," *IEEE Communications Letters*, vol. 5, no. 2, pp. 58–60, Feb 2001.
- [5] R. Cohen and Y. Cassuto, "LDPC codes for partial-erasure channels in multi-level memories," *2014 IEEE International Symposium on Information Theory*, pp. 2087–2091, July 2014.
- [6] ———, "Estimation of the decoding threshold of LDPC codes over the q -ary partial erasure channel," *8th International Symposium on Turbo Codes & Iterative Information Processing*, pp. 198–202, August 2014.
- [7] R. M. Tanner, "A recursive approach to low complexity codes," *IEEE Trans. Inform. Theory*, vol. IT-27, pp. 533–547, 1981.
- [8] T. Richardson and R. Urbanke, "The capacity of low-density parity-check codes under message-passing decoding," *IEEE Transactions on Information Theory*, vol. 47, no. 2, pp. 599–618, 2001.
- [9] A. Bennatan and D. Burshtein, "Design and analysis of nonbinary LDPC codes for arbitrary discrete-memoryless channels," *IEEE Transactions on Information Theory*, vol. 52, no. 2, pp. 549–583, 2006.
- [10] T. Richardson, A. Shokrollahi, and R. Urbanke, "Design of capacity-approaching irregular low-density parity-check codes," *IEEE Transactions on Information Theory*, vol. 47, no. 2, pp. 619–637, 2001.