Quantization of Binary-Input Discrete Memoryless Channels, with Applications to LDPC Decoding

Brian M. Kurkoski and Hideki Yagi

Abstract—The quantization of the output of a binary-input discrete memoryless channel to a smaller number of levels is considered. The optimal quantizer, in the sense of maximizing mutual information between the channel input and the quantizer output, may be found by an algorithm with complexity which is quadratic in the number of channel outputs. This is a concave optimization problem, and results from the field of concave optimization are invoked. The quantizer design algorithm is a realization of a dynamic program.

Then, this algorithm is applied to the design of message-passing decoders for low-density parity-check codes, over arbitrary discrete memoryless channels. A general, systematic method to find message-passing decoding maps which maximize mutual information at each iteration is given. This may contrasted with existing quantized message-passing algorithms which are heuristically derived. The method finds message-passing decoding maps similar to those given by Richardson and Urbanke's Algorithm E. Using four bits per message, noise thresholds similar to belief-propagation decoding are obtained.

Index Terms—discrete memoryless channel, channel quantization, mutual information maximization, LDPC decoding

I. INTRODUCTION

The problem of finding good channel quantizers is of importance since most communications receivers convert physical-world analog values to discrete values. It is these discrete values that are used by subsequent filtering, detection and decoding algorithms. Since the complexity of circuits which implement such algorithms increases with the number of quantization levels, it is desirable to use as few levels as possible, for some specified error performance.

Channel capacity is the maximization of mutual information, so a reasonable metric for designing channel quantizers is to similarly maximize mutual information between the channel input and the quantizer output. For a memoryless channel with a fixed input distribution, the quantizer which maximizes mutual information will give the highest achievable communications rate. Previous work on quantization with a mutual information objective function has concentrated on continuous-to-discrete quantization with “locally optimal” algorithms, as will be explained in the following section.

This paper gives two results for channels with discrete, rather than continuous, alphabets. The first result is a quadratic-complexity algorithm that finds the quantizer for a binary-input discrete memoryless channel (DMC) which globally maximizes the mutual information between the channel input and the quantizer output. The second result is the application of this algorithm to find message-passing decoding mappings for low-density parity-check (LDPC) codes which maximize mutual information. Both results apply for arbitrary binary-input DMCs.

The first result can be concisely stated as follows: consider a DMC with inputs $X$ and outputs $Y$, as shown in Fig. 1. A quantizer $Q$ maps $I$ channel outputs to $K$ quantizer outputs $Z$, with $K < I$ for cases of interest. The set of all possible quantizers is denoted by $Q$. Under these conditions, the following Theorem for arbitrary DMCs holds:

**Theorem.** The quantizer $Q^*$ which maximizes the mutual information between $X$ and $Z$:

$$Q^* = \arg \max_{Q \in Q} I(X; Z),$$  \hspace{1cm} (1)

can be found with complexity proportional to $I^2$, when $X$ is binary.

This maximization (1) is a concave optimization problem, and this paper invokes results from the field of concave optimization. Concave optimization is NP-hard in general, and a naive approach requires complexity exponential in $I$. The significance of the Theorem is that the optimal quantizer may be found with complexity quadratic in $I$. This problem

---

The authors are with the Dept. of Information and Communication Engineering, University of Electro-Communications, Tokyo, Japan; email kurkoski@ice.uec.ac.jp and yagi@ice.uec.ac.jp.

Part of this work was previously presented at the 2010 Information Theory Workshop (ITW) [1] and at the 2008 Global Communications Conference (Globecom) [2].

B.K. was supported in part by the Ministry of Education, Science, Sports and Culture; Grant-in-Aid for Scientific Research (C) number 21560388 and Grant-in-Aid for Scientific Research (C) number 23560439. H.Y. was supported in part by the Ministry of Education, Science, Sports and Culture; Grant-in-Aid for Young Scientists (B) number 22700270 and IST’s Special Coordination Funds for Promoting Science and Technology.

Submitted to Transactions on Information Theory on July 28, 2011.
superficially resembles several familiar information-theoretic problems, such as determination of the rate-distortion function. However, as discussed in the following section, these problems are distinct because they are convex optimization problems.

This optimization problem is formulated as a dynamic program, which results in a quantizer design algorithm, which is referred to as the Quantization Algorithm. While the algorithm assuming uniformly-distributed inputs appeared in conference proceedings previously [1], contributions of this paper include the proof of its optimality and an extension to non-uniform input distributions. A necessary condition for quantizer optimality, to be given as Lemma 3, is used to establish this result.

The second result of this paper is to demonstrate the utility of the Theorem, by applying the Quantization Algorithm to the design of message-passing LDPC decoders with a fixed message alphabet size. Specifically, this paper describes a systematic method to design LDPC decoder message quantization and message-passing decoding maps for an arbitrary DMC and an arbitrary number of quantization levels. This method can be viewed from several perspectives: it produces non-uniform quantization of decoder messages, it produces message-passing decoding maps which maximize mutual information, and it compresses the messages.

The method applies the Quantization Algorithm at each step of density evolution; the obtained quantizers are then used to generate message-passing decoding maps. These maps may then be used to implement a finite-length decoder. They do not necessarily correspond to mathematical operations, but may be implemented by a look-up table. Numerical results show that using four bits per message gives error-rate performance similar to full belief-propagation decoding. In addition, the maps that this method automatically produces are similar to heuristically-derived decoding algorithms, such as Algorithm E [3].

The flow of the remainder of this paper is as follows. Related work is reviewed in Sec. II, and its relationship with this paper is stated. Sec. III (a) formalizes the problem statement as a concave programming problem (b) shows that the optimal quantizer is deterministic and (c) gives Lemma 3, a necessary condition on the optimality of the quantizer. Then, Sec. IV gives the Quantization Algorithm, which implements a dynamic programming approach to find the optimal quantizer.

In Sec. V, the method for quantizing LDPC decoder messages and finding the associated message-passing decoding maps is described. Then, numerical results are reported in Sec. VI, showing that noise thresholds for a quantized LDPC decoder can be quite close to those for floating-point decoding. Some discussion is given in Sec. VII. Proofs and additional lemmas are in the Appendix.

II. RELATIONSHIP WITH PRIOR WORK

This section establishes the relationship between the problem studied in this paper and previous results. To do so, first notation is briefly given. Then, the relationship with well-known information extremum problems is described. After that, an overview of work on information-theoretic channel quantizer design is given. Finally, a sampling of related work on LDPC message quantization is given.

A. Notation

Here, the notation of Sec. I is formalized. The alphabet sizes of $X$, $Y$ and $Z$ are $J$, $I$ and $K$, respectively. Define as follows:

$$p_j = \Pr(X = j),$$

with $j = 1, \ldots, J$,

$$r_i = \Pr(Y = i),$$

with $i = 1, \ldots, I$,

$$q_k = \Pr(Z = k),$$

with $k = 1, \ldots, K$,

$$P_{i|j} = \Pr(Y = i|X = j),$$

$$Q_{k|i} = \Pr(Z = k|Y = i),$$

and

$$T_{k|ij} = \Pr(Z = k|X = j) = \sum_i Q_{k|i} P_{i|j}.$$  

Except for Lemma 2 (given in the next section), this paper makes the restriction that $J = 2$. The sum $\sum_i$, etc. means the sum over the whole alphabet $\sum_{i=1}^{I}$, etc.

The mutual information between two variables, say $X$ and $Z$, is:

$$I(X; Z) = \sum_k \sum_{j} p_j T_{k|ij} \log \frac{T_{k|ij}}{p_j \sum_{j'} T_{k|ij'}}.$$  

(2)

It is well-known that mutual information is convex (lower convex) in $T_{k|ij}$, for fixed $p_j$. Similarly, it is concave (upper convex) in $p_j$ for fixed $T_{k|ij}$ [4, Theorem 2.7.4].

The quantizer is given by $Q_{k|i}$, $k = 1, \ldots, K$ and $i = 1, \ldots, I$ and may be regarded as a $K \times I$ matrix. The set of possible quantizers $Q$ consists of all $K \times I$ stochastic matrices, that is, with

$$\Pi = \left\{ \pi_i \mid \sum_{i=1}^{K} \pi_i = 1, \pi_i \geq 0 \right\},$$  

(3)

then

$$Q^* = \Pi^\dagger.$$  

(4)

B. Superficially Related Problems

Superficially, the information maximization problem, restated here for convenience,

$$Q^* = \arg \max_{Q \in \Pi} I(X; Z),$$  

(5)
appears similar to various information-theoretic optimization problems, particularly the computation of the rate-distortion function,

\[ R(D) = \min_{Q \in \mathcal{Q}} I(Y; Z), \]  

but it is distinct. Mutual information is convex in \( Q \), and for the computation of \( R(D) \), mutual information is minimized, so this is a convex optimization problem [4, Sec. 13.7]. On the other hand, the channel quantization problem is to maximize mutual information in \( Q \), leading to a considerably different concave optimization problem. Although \( X \) was replaced with \( Y \) to make the comparison, as discussed in the following section, this affine transform does not change the convexity of mutual information. Also, the distortion constraint was omitted for clarity.

For the computation of the DMC capacity, mutual information is maximized over the input distribution \( p_j \), which is concave, while the channel transition probabilities are fixed. For problem (1), the maximization is in \( Q_{k|i} \) while the \( p_j \) are fixed; this is also distinct. The relationship between the rate-distortion problem, the DMC capacity, and the problem treated in this paper is illustrated in Fig. 2. The well-known algorithm to compute the channel capacity and rate-distortion function, given by Arimoto [5] and Blahut [6], is a convex optimization method.

Another information extremum problem is the information bottleneck method, from the field of machine learning [7]. The problem setup is identical, using the same Markov chain \( X \to Y \to Z \). However, this is an information minimization problem, using a Lagrange multiplier to sweep a kind of rate-distortion curve. Moreover, it is a convex optimization method, using alternating minimization.

C. Information-Theoretic Quantizer Design Criteria

The channel cutoff rate, another information theoretic measure, was suggested as a criterion for designing quantizers for continuous-output channels, by Wozencraft and Kennedy in the 1960s [8]; see also [9, Sec. 6.2]. Massey went on to emphasize that the cutoff rate was superior to the probability of error as a receiver design criterion, and gave an algorithm to find a channel quantizer which maximizes the cutoff rate for the binary-input AWGN channel [10]. Lee extended these results to channels with non-binary inputs, and gave an algorithm to find decision regions for continuous-output channels [11]. If the metrics for each output are restricted to integers, this can reduce the decoder complexity, and again can be designed using the cutoff rate [12]. The channel cutoff rate has been used to quantize turbo decoders, as well [13].

While the cutoff rate is an important information theoretic measure, since the channel capacity, which is above the cutoff rate, can now be practically approached with LDPC codes, mutual information is a more appropriate measure. Also, the restriction to integer metrics is practical when decoding operations such as additions and multiplications are used; however, the LDPC decoder described in this paper does not use mathematical operations, instead decoding mappings could be implemented by lookup tables.

More recently, maximization of mutual information has been considered as a criterion for channel quantizers. Most papers emphasize continuous-to-discrete quantization of the AWGN channel. The earliest work we are aware of is the 2002 conference paper of Ma et al., which considered quantization of the binary-input AWGN channel [14]. For the special case of three quantizer outputs, it is straightforward to select a single parameter which maximizes mutual information. However, for a larger number of outputs, local optimization is feasible and this has higher mutual information than uniform quantization [15]. Singh et al. considered the problem of jointly finding capacity-achieving input distributions and AWGN channel quantizers [16]. Again, for an AWGN channel quantized to 3 levels, optimization over a single parameter was done, but for a larger number of outputs, a local optimization algorithm was used. In fact, this is a concave-convex problem, and global optimization appears difficult.

While these locally optimal quantization algorithms appear to be effective, the subject of this paper is a globally optimal quantization. Also, whereas previous work concentrated on continuous output (and usually symmetrical) channels, the results here are obtained by a different approach: by working with quantized channels, rather than working directly with continuous output channels. Of course, a continuous output channel can be approximated with arbitrarily small discrepancy by an finely quantized channel. While previous papers concentrated on AWGN channels, the results in this paper hold for arbitrary DMCs. Thus, we believe that this is the first result on globally optimal quantization of general binary-input channels.

In addition, as far as we are aware, all previous work \textit{a priori} assumed deterministic quantizers were optimal. In this paper, it is shown that this assumption is valid, in Lemma 2.

D. Design of Quantized LDPC Decoders

The quantization of LDPC decoder messages is of great practical importance, and has received substantial attention in both the communication theory and VLSI literature. Belief-propagation decoding of LDPC codes uses real numbers for decoder messages, but most real-world implementations must quantize these real numbers. Accordingly, substantial attention has been directed at the design of quantized LDPC decoders. See [17] for a recent example.

An efficient decoding approach is to use a bit-flipping algorithm, such as Gallager B for the binary symmetric channel, which uses only one bit per message. There are numerous variations on bit-flipping decoding for continuous-output channels [18]. But using one bit per message has a performance penalty. Algorithm E uses three-level decoder messages (0, 1 and erasure) to decode LDPC codes transmitted over the binary symmetric channel (BSC). The algorithm assigns an iteration-dependent weighting factor to the channel value, selected by maximizing mutual information [3]. Using two bits per message can further improve performance while being suitable for analysis and decoding on graphs with cycles [19]. Quantized LDPC belief-propagation decoders can be designed by considering mutual information, resulting in non-uniform message quantization, and it has been shown that
using four bits per message is quite close to unquantized performance [20]. This is significant since conventional uniform quantization requires about six bits per message to achieve similar error performance.

The LDPC decoding method described in this paper is in some sense a generalization of these previous works, where reasonable heuristics were used to specify the message-passing decoding rules, sometimes considering mutual information. But in this paper, no assumptions are made about the decoding rules. Instead, maximization of mutual information and the number of quantization levels are the only design criteria, and the specific message-passing decoding maps are found using the Quantization Algorithm, presented in Sec. IV. The method is similar to the one previously presented [2], although a greedy algorithm was used for quantization. The distinctions of this paper include the use of the efficient Quantization Algorithm and the handling of asymmetric channels.

III. CONCAVE OPTIMIZATION AND A NECESSARY CONDITION FOR OPTIMALITY

A. Concave Optimization

Concave optimization, also known as concave programming or concave minimization, is a class of mathematical programming problems which has the general form:

\[
\min f(x), \text{ subject to } x \in S, \quad (7)
\]

where \( S \subseteq \mathbb{R}^n \) is a feasible region and \( f(x) \) is a concave function [21] [22]. There is substantial literature on concave optimization, and it is generally considered to be computationally more difficult than convex optimization. General concave optimization is NP-hard; more efficient methods may be found, but are problem dependent, as is the case in this paper. In concave optimization, there are multiple local minima, which is distinct from convex optimization. The following is an important result from the field of concave optimization that is the feasible region.

\[
\text{Lemma 1. [22, Theorem 1.19]} \text{ A concave (convex) function } f: S \rightarrow \mathbb{R} \text{ attains its global minimum (maximum) over } S \text{ at an extreme point of } S.
\]

The literature on concave optimization uses minimization of a concave function, but in the remainder of this paper, the equivalent maximization of a convex function is used.

If \( S \) is a polytope, as in this paper, then the extreme points are its vertices. Lemma 1 can be visualized in one dimension in Fig. 2-(a), where it is clear that the maximum must be at either endpoint 0 or 1, that is, the vertices of the line segment that is the feasible region.

The objective of the maximization in (1) is given by:

\[
\max \sum_{k} \sum_{j} p_j \sum_{i} Q_{k|j} P_{i|j} \log \frac{\sum_{i'} Q_{k|i'} P_{i'|j}}{\sum_{i'} Q_{k|i'} P_{i'|j}'}, \quad (9)
\]

subject to:

\[
\sum_{i} Q_{k|i} = 1, \quad i = 1, \ldots, I \quad \text{and} \quad Q_{k|i} \geq 0, \quad i = 1, \ldots, I \quad \text{and} \quad k = 1, \ldots, K
\]

The constraint enforces that \( Q_{k|i} \) is a conditional probability distribution.

B. The Optimal Quantizer Is Deterministic

This section demonstrates the following:

\[
\text{Lemma 2. For any DMC and any } K, \text{ the optimal quantizer } Q^* \text{ is deterministic. That is, } Q_{k|i} \in \{0, 1\}, \text{ for all } i \text{ and } k.
\]

The feasible region in (9) is a polytope. According to Lemma 1, the optimal quantizer attains its global maximum at one of the polytope vertices. To prove Lemma 2, it is sufficient to show that the extreme points, or vertices, are at \( Q_{k|i} = 0 \) or 1.

Consider the portion of the polytope for some fixed \( i \). The vertices of this part of the feasible region are where the hyperplane \( \sum_k Q_{k|i} = 1 \) intersects the \( K \) half-spaces \( Q_{k|i} \geq 0 \). These \( K \) vertices are:

\[
\{ (1, 0, \ldots, 0), \quad (0, 1, \ldots, 0), \quad \ldots, \quad (0, 0, \ldots, 1) \}.
\]

For each \( i = 1, \ldots, I \), these \( K \) vertices may be selected independently and all vertices are at 0 or 1. By Lemma 1, the solution is at one of the \( K^* \) vertices, which completes the proof. Note that Lemma 2 holds for an arbitrary number of inputs \( J \).

As an example to illustrate Lemma 2, consider the binary, symmetric errors and erasure channel, with the transition matrix:

\[
P = \begin{bmatrix} 1-p-q & p & q \\ q & p & 1-p-q \end{bmatrix},
\]

for \( p, q \geq 0 \) and \( p + q \leq 1 \). Suppose the three outputs, called 0, erasure and 1, are to be quantized to two levels. One might expect that symmetry should be maintained by mapping the erasure symbol to the two output symbols with probability 0.5 each. However, as Lemma 2 shows, this probabilistic assignment has lower mutual information than mapping the erasure symbol to either 0 or 1 with probably one. This optimal quantizer lacks symmetry between the channel input and quantizer output.
A naive optimization approach for (9) would be to search over all $K^I$ candidate solutions, which is searching over all deterministic quantizers. This has complexity which is exponential in $I$. However, as will be shown in the sequel, a more efficient algorithm exists.

C. Necessary Condition for Optimality

From Lemma 2, for each channel output $i$, there is exactly one value $k'$, for which $Q_{k'|i} = 1$, and for all other values of $k$, $Q_{k|i} = 0$. For a given quantizer output $k$, let $A_k$ be the set of values $i$ for which $Q_{k|i} = 1$. The quantizer is a mapping from channel outputs to quantizer outputs. Under this mapping, $A_k$ is the preimage of $k$. The sets $A_m$ and $A_n$ are disjoint for $m \neq n$, and the union of all the sets is $\{1, 2, \ldots, I\}$.

Lemma 3 describes a necessary condition for the quantizer to be optimal. It is key for proving that the Quantization Algorithm produces the optimal quantizer. The proof of Lemma 3 is given in the Appendix.

**Lemma 3.** For the quantizer which maximizes mutual information, the preimage $A_k$ consists of consecutive channel outputs, for each $k = 1, \ldots, K$, when the channel outputs are sorted according to:

\[
\frac{P_{i|1}}{P_{i|2}} < \frac{P_{i+1|1}}{P_{i+1|2}} < \cdots < \frac{P_{T-1|1}}{P_{T-1|2}} < \frac{P_{T|1}}{P_{T|2}},
\]

(11)

Note that there is no loss of generality because the outputs can always be re-labeled such that (11) holds. In addition, the log-likelihood ratio for channel output $i$ is $\log \frac{P_{i|1}}{P_{i|2}}$. Since the log function is monotonically increasing, the sorting condition is equivalent to sorting the log-likelihood ratios.

Strict inequalities are used in (11) because if

\[
\frac{P_{i|1}}{P_{i|2}} = \frac{P_{i+1|1}}{P_{i+1|2}},
\]

then outputs $i$ and $i+1$ can be combined to a single output with the likelihood $P_{i|j} + P_{i+1|j}$ for input $j$ to form a new channel with $I-1$ outputs. The likelihood ratio for the combined output,

\[
\frac{P_{i|1} + P_{i+1|1}}{P_{i|2} + P_{i+1|2}}
\]

is equal to (12). In addition, the original channel and the new channel have the same mutual information. The use of strict inequalities simplifies the proofs in the Appendix.

**Remark 1.** One might expect that the sorting condition (11) should depend upon the input distribution $p_j$, but it is independent. In fact, the sorting condition can be rewritten as:

\[
\frac{p_1 P_{1|1}}{p_2 P_{1|2}} < \frac{p_1 P_{2|1}}{p_2 P_{2|2}} < \cdots < \frac{p_1 P_{T-1|1}}{p_2 P_{T-1|2}} < \frac{p_1 P_{T|1}}{p_2 P_{T|2}},
\]

which is equivalent to (11). Thus, the condition introduced for uniformly-distributed inputs previously [1], is valid for arbitrary input distributions as well.

Thus, interest may be restricted to the cases where, $A_1$ is the set $\{1, 2, \ldots, a_1\}$, and $A_2$ is the set $\{a_1 + 1, \ldots, a_2\}$, etc., and $A_K$ is the set $\{a_{K-1} + 1, \ldots, I\}$, with,

\[
1 \leq a_1 < a_2 < \cdots < a_{K-1} < I.
\]

(15)

Each $A_k$ has at least one element. For convenience, let $a_0 = 0$ and $a_K = I$. While there are $K^I$ possible deterministic quantizers, the optimal quantizer is contained in a subset, and the Quantization Algorithm searches this subset.

IV. QUANTIZATION ALGORITHM

Lemma 3 states that a necessary condition for a quantizer optimality is that the preimage of a quantizer output are consecutive channel outputs. For a given channel, finding the optimal quantizer reduces to finding the boundaries $a_1', a_2', \ldots, a_{K-1}'$ which maximize the mutual information. This section describes an algorithm which finds those boundaries. First, partial mutual information is described.

**A. Partial Mutual Information**

A partial sum of mutual information, called “partial mutual information,” is used both in the Quantization Algorithm and in the proofs in the Appendix. Partial mutual information $\iota$ is the contribution that one or more quantizer outputs makes to the total mutual information. For a deterministic quantizer, the total mutual information is:

\[
I(X; Z) = \sum_k \sum_j p_j \sum_{i \in A_k} \log \frac{\sum_{i' \in A_k} P_{i'|j} \sum_{i'' \in A_k} P_{i'' |j'}}{\sum_{i'' \in A_k} \sum_{i' \in A_k} P_{i'' |j'}},
\]

since $Q_{k|i} = 1$ if and only if $i \in A_k$.

Under the quantization mapping from channel outputs to quantizer outputs, the preimage of quantizer output $m$ is $A_m$. The partial mutual information $\iota_m$ for this output is:

\[
\iota_m = \sum_j p_j \sum_{i \in A_m} \log \frac{\sum_{i' \in A_m} P_{i'|j} \sum_{i'' \in A_m} P_{i'' |j'}}{\sum_{i'' \in A_m} \sum_{i' \in A_m} P_{i'' |j'}},
\]

(16)

so total mutual information is the sum of all the partial mutual information terms:

\[
I(X; Z) = \sum_k \iota_k.
\]

(17)

Further, let consecutive channel outputs $a' + 1$ to $a$, with $a' < a \leq I$, be assigned to a single quantizer output. Denote by $\iota(a' \rightarrow a)$, the partial mutual information:

\[
\iota(a' \rightarrow a) = \sum_j p_j \sum_{i = a' + 1}^{a} \log \frac{\sum_{i' = a' + 1}^{a} P_{i'|j} \sum_{i'' = a' + 1}^{a} P_{i'' |j'}}{\sum_{i'' = a' + 1}^{a} \sum_{i' = a' + 1}^{a} P_{i'' |j'}},
\]

(18)

So if $A_k = \{a_{k-1} + 1, \ldots, a_k\}$, then $\iota_k = \iota(a_{k-1} \rightarrow a_k)$.

**B. Quantization Algorithm**

The Quantization Algorithm is a quantizer design algorithm and is the realization of a dynamic program. The algorithm has a state value $S_k(i)$, which is the maximum partial mutual information when channel outputs 1 to $i$ are quantized to
quantizer outputs 1 to \( k \). This can be computed recursively by conditioning on the state value at time index \( k - 1 \):

\[
S_k(a) = \max_{a'} \left( S_{k-1}(a') + \iota(a' \to a) \right), \quad (19)
\]

where the maximization is taken over \( a' \in \{0, 1, \ldots, a-1\} \).

3) Precompute partial mutual information. For each \( a' \in \{1, \ldots, I-1\} \) and for each \( a \in \{a'+1, \ldots, I\} \) (where \( t = \min(\{a'+1 + I - K, I\}) \)):
   - compute \( \iota(a' \to a) \) according to (18).
4) Recursion. For each \( k \in \{1, \ldots, K\} \), and for each \( a \in \{k, \ldots, k + I - K\} \),
   - compute \( S_k(a) \) according to (19),
   - store the local decision \( h_k(a) \):
     \[
     h_k(a) = \arg\max_{a'} S_{k-1}(a') + \iota(a' \to a),
     \]
   - where the maximization is taken over \( a' \in \{k-1, \ldots, a-1\} \).
5) Find the optimal quantizer by traceback. Let \( a^*_k = I \).
   - For each \( k \in \{K-1, K-2, \ldots, 1\} \):
     \[
     a^*_k = h_{k+1}(a^*_{k+1}).
     \]

Fig. 4: Optimal quantization to \( K = 8 \) levels of a DMC derived from a finely-quantized binary-input AWGN channel, with noise variance \( \sigma^2 \). Solid and dotted lines show quantization boundaries when the AWGN channel is quantized to \( I = 500 \) and \( I = 30 \) levels, respectively.

6) Outputs:
   - The optimal \( a_1^*, a_2^*, \ldots, a_{K-1}^* \). Equivalently, output the matrix \( Q^* \), where row \( k \) of \( Q^* \) has ones in columns \( a_{k-1} + 1 \) to \( a_k \) and zeros in all other columns.
   - The maximum mutual information, \( S_K(I) \).

There may be multiple optimal quantizers. In an implementation, storage of the local decision and traceback should deal with ties. This was not explicitly indicated, to keep the notation simple.

The main computational burden is to pre-compute \( \iota(a' \to a) \) in step 3. Since \( a' \) is from a set of size \( I \) and \( a \) is from a set of size at most \( I - K + 1 \), the number of \( \iota \) computations is proportional to \( I^2 \). Note that in (18), the sum on \( i \) could be over as many as \( I - K + 1 \) terms. However, since this sum can be computed recursively, the complexity remains proportional to \( I^2 \).

Also, for each \( k \) in step 4, roughly \( \frac{1}{2} (I - K)^2 \) add/compare operations are needed, and there are \( K \) such steps. This results in number of operations roughly \( \frac{1}{2} K (I - K)^2 \). This is also proportional to \( I^2 \). Note that if \( I \) is not much larger than \( K \), then \( (I - K) \) is close to zero. Thus, the computational complexity is quadratic in \( I \).

This complexity result, along with the proof of optimality in the Appendix, proves the Theorem.

Source code which implements the Quantization Algorithm, as well as the density evolution method in the next section, is available [24].
C. Finely Quantized Continuous-Output Channel

While the Quantization Algorithm can only be applied to discrete channels, it can be used to obtain good coarse quantization of a continuous-output channel, by first using fine quantization. This can be illustrated for the binary-input AWGN channel with ±1 inputs and Gaussian noise variance \( \sigma^2 \). Fig. 4 was created by first uniformly quantizing the channel between \(-2\) and \(+2\) with \( I = 500 \) or \( I = 30 \) steps. Then, the Quantization Algorithm is applied with \( K = 8 \) quantizer outputs. The figure shows the \( K-1 = 7 \) quantization boundaries when the AWGN channel was finely quantized with \( I = 500 \) (solid line) and \( I = 30 \) (dashed line).

V. APPLICATION TO DECODING LDPC CODES

A. Overview

In this section, the Quantization Algorithm is used to find decoding mappings for LDPC decoders. Alternatively, this may be viewed as finding optimum non-uniform quantization, or as efficient message compression.

No assumptions are made about the message-passing maps. Instead, maps are derived from the quantizer generated by the algorithm. As a result, the decoding maps maximize mutual information. The “channel” being quantized is not a model of a physical communication system, but a conditional probability density on the LDPC decoder messages. In particular, the LDPC code bits will play the role of \( X \) and the decoder messages will play the role of \( Z \). A cross-product distribution will play the role of \( Y \).

The maps are generated in the context of density evolution [3], and an overview of this method is shown in Fig. 5. In density evolution, at each degree-\( d \) node and each iteration, a given conditional input distribution is used to determine an output distribution. Each of \( d - 1 \) incoming messages is a discrete distribution with \( K \) values. An intermediate (cross-product) message distribution with \( K^{d-1} \) values is created, and is simply the cross-product of the incoming message distributions. The key step comes when we apply the Quantization Algorithm to this cross-product distribution, which produces a quantizer that reduces the distribution to \( K \) values. This quantizer is then used to find the message-passing decoding map. These decoding mappings are locally optimal, in the sense of maximizing mutual information at each decoding iteration; we cannot say anything about global optimality over all iterations. Also, by using density evolution, finite-length effects can be ignored, and only the channel, the LDPC node degrees and the value \( K \) are required as inputs to this procedure.

Classical density evolution is restricted to channels with certain symmetry properties [3]. But here, arbitrary and asymmetrical channels are allowed, and the optimized decoding maps, and thus the distributions, may be asymmetrical even if the channel was symmetrical. Fortunately, Wang et al. generalized density evolution to asymmetric channels [25]. They showed that while error rates are codeword-dependent, it is sufficient to consider the evolution of densities only for the two code bits, that is densities conditioned on \( X = 0 \) and \( X = 1 \). The same method will be used here.

B. Message-Passing Decoding of LDPC Codes

An arbitrary, binary-input DMC is used for transmission; it has binary inputs \( X \) and \( K_{ch} \) outputs \( W \in W = \{1, 2, \ldots, K_{ch}\} \). The channel transition probabilities are denoted by \( r^{(0)} \):

\[
r^{(0)}(x_0, y_0) = \Pr(W = y_0 | X = x_0).
\]

In message-passing decoding [3], the variable-to-check messages \( R \) are from the message alphabet \( R \). Similarly, the check-to-variable messages \( L \) are from the message alphabet \( L \). The sets \( R \) and \( L \) are discrete with \( |R| \leq K \) and \( |L| \leq K \).

At iteration \( \ell \), the check node with degree \( d_c \) finds an outgoing message using \( d_c - 1 \) incoming messages, by a mapping function:

\[
\chi^{(\ell)} : \mathcal{R}^{d_c-1} \rightarrow \mathcal{L}.
\]

Similarly, at iteration \( \ell \), the variable node with degree \( d_v \) finds an outgoing message using the channel value and \( d_v - 1 \) incoming messages, by a mapping function:

\[
\Phi^{(\ell)} : \mathcal{W} \times \mathcal{L}^{d_v-1} \rightarrow \mathcal{R}.
\]

C. Density Evolution

The object of interest is the density of the messages \( R \) and \( L \). Because of possible asymmetries, density evolution tracks the probabilities conditioned on both \( X = 0 \) and \( X = 1 \). On iteration \( \ell \), the probability distribution for \( R \) is:

\[
r^{(\ell)}(x, y) = \Pr(R = y | X = x),
\]

with \( y \in \mathcal{R} \), and the probability distribution for \( L \) is:

\[
l^{(\ell)}(x, y) = \Pr(L = y | X = x),
\]

with \( y \in \mathcal{L} \).

The method described here finds the message-passing decoding maps \( \chi \) and \( \Phi \), as well as the probability distributions \( r \) and \( l \). In particular, for each iteration and each node type, there are four steps: (a) given the node input distribution, a cross-product distribution is found; (b) the Quantization Algorithm
produces a quantizer to $K$ levels; (c) the reduced distribution is found, which is used in the next step of density evolution; 
(d) the decoding mapping is found for each quantizer.

Notation is given first. Two functions $f_c$ and $f_v$ are of interest when decoding LDPC codes. At the check node:

$$f_c(x_1, \ldots, x_{d_c-1}) = x_1 + \cdots + x_{d_c-1} \mod 2 \quad (25)$$

and at the variable node:

$$f_v(x_0, \ldots, x_{d_v-1}) = \begin{cases} 
0 & \text{if } x_0 = x_1 = \cdots = 0 \\
1 & \text{if } x_0 = x_1 = \cdots = 1 \\
\text{otherwise undefined} & \end{cases} \quad (26)$$

where $x_i$ are binary values. It is convenient to use a single symbol that is a concatenation of the component messages in the cross-product distribution. In the context of the check node, let $y'$ denote the concatenation:

$$y' = (y_1, y_2, \ldots, y_{d_c-1}), \quad (27)$$

where $y' \in \mathcal{R}_1^{d_c-1}$. And in the context of the variable node, let $y'$ denote the concatenation:

$$y' = (y_0, y_1, y_2, \ldots, y_{d_v-1}), \quad (28)$$

where $y' \in \mathcal{W} \times \mathcal{L}^{d_v-1}$.

Step (a) is to find the cross-product distributions $\bar{\ell}(\ell)(x, y')$ and $\bar{\mu}(\ell)(x, y')$, given by:

$$\bar{\ell}(\ell)(x, y') = \frac{1}{2} \sum_{x: f_i(x)=x} \prod_{i=1}^{d_c-1} \ell^{(\ell-1)}(x_i, y_i), \quad (29)$$

where $x = (x_1, x_2, \ldots, x_{d_c-1})$, and

$$\bar{\mu}(\ell)(x, y') = \sum_{x: f_i(x)=x} \mu^{(0)}(x_0, y_0) \prod_{i=1}^{d_v-1} \ell^{(\ell)}(x_i, y_i), \quad (30)$$

where $x = (x_0, x_1, \ldots, x_{d_v-1})$.

Then, in step (b), the cross-product distribution is reduced to $K$ levels using the Quantization Algorithm. The matrix-form quantizers $Q_c^{(\ell)}$ and $Q_v^{(\ell)}$ are produced at each iteration $\ell$, given by:

$$Q_c^{(\ell)} = \text{Quant}(\bar{\ell}(\ell), K) \quad (31)$$

$$Q_v^{(\ell)} = \text{Quant}(\bar{\mu}(\ell), K). \quad (32)$$

Since LDPC codes are linear codes, code bits 0 and 1 occur with equal probability. Thus the input distribution $p_1 = p_2 = 0.5$ is used by the Quantization Algorithm.

Step (c) is to find the reduced distributions as:

$$l^{(\ell)}(x, y) = \sum_{y': Q_c^{(\ell)}(y, y')=1} \bar{\ell}(\ell)(x, y') \quad (33)$$

$$r^{(\ell)}(x, y) = \sum_{y': Q_v^{(\ell)}(y, y')=1} \bar{\mu}(\ell)(x, y'), \quad (34)$$

where $Q(y, y')$ is the element in row $y$ and column $y'$ of the matrix $Q$.

Step (d) is to find the decoding maps, which are given by:

$$\chi^{(\ell)}(y') = y \quad \text{if } Q_c^{(\ell)}(y, y') = 1, \quad (35)$$

$$\Phi^{(\ell)}(y') = y \quad \text{if } Q_v^{(\ell)}(y, y') = 1. \quad (36)$$

Since each column of $Q_c^{(\ell)}$ and $Q_v^{(\ell)}$ has a single 1, the functions $\chi^{(\ell)}(y')$ and $\Phi^{(\ell)}(y')$ are defined for all values of $y'$. This step is not needed for density evolution.

More precisely, density evolution is as follows:

1. Initialize with $\ell = 0$ and the channel message given by (20).
2. Check node: Compute (29), followed by (31), followed by (33).
3. Variable node: Compute (30), followed by (32), followed by (34).
4. If the mutual information $I(X; R)$ approaches 1, then declare convergence. If a fixed number of iterations is exceeded, declare non-convergence. Otherwise, increment $\ell$ and iterate from the check node step.

At steps 2 and 3 above, decoding mappings can also be obtained.

Density evolution steps (a), (b) and (c) may be schematically represented as the following flow:

$$r^{(0)} \rightarrow \bar{l}^{(1)} \rightarrow Q_c^{(1)} \rightarrow l^{(1)} \rightarrow \bar{r}^{(2)} \rightarrow Q_v^{(2)} \rightarrow r^{(2)} \rightarrow \bar{r}^{(3)} \rightarrow Q_v^{(3)} \rightarrow r^{(3)} \rightarrow \ldots$$

Also, step (d) finding the decoding mapping can be schematically represented as:

$$Q_c^{(1)} \rightarrow \chi^{(1)}$$
$$Q_v^{(1)} \rightarrow \Phi^{(1)}$$
$$Q_c^{(2)} \rightarrow \chi^{(2)}$$
$$Q_v^{(2)} \rightarrow \Phi^{(2)}$$
$$Q_c^{(3)} \rightarrow \chi^{(3)}$$

$$\vdots$$

Decoding can be implemented as follows: on the first iteration, the channel messages $W$ are sent to the check node which uses the mapping $\chi^{(1)}$ to generate check-to-variable messages $L$. Using messages $L$ and $W$, the variable node uses the mapping $\Phi^{(1)}$ to generate variable-to-check messages $R$. This continues iteratively, until a stopping condition, such as convergence to a codeword or maximum number of iterations, is satisfied.

Note that the Quantization Algorithm may also be used to design a mapping function which makes a hard decision on $x$. For each iteration, repeat the variable node steps (a) to (c), using all $d_v$ inputs and quantize to $K = 2$ levels to make hard decisions.

The complexity of this density evolution method is dominated by the Quantization Algorithm at step (b). The complexity is proportional to $K^2$. For density evolution, $I = K^d$ for a degree $d$ variable node and $I = K^{d-1}$ for a degree $d$ check node. Thus, we have complexity proportional to roughly $I^{2+1/d}$, which is slightly worse than quadratic.
VI. NUMERICAL RESULTS FOR LDPC CODES

Using the method of the previous section, this section gives noise thresholds for LDPC decoders with quantized messages, on quantized AWGN channels. Also, examples of decoding mappings $\chi$ and $\Phi$ are given.

A. Noise Thresholds for AWGN Channel

Because the complexity of the method given in the previous section is comparatively low, it can be effectively used as a way to compare quantization various schemes.

For the numerical evaluation, a binary-input AWGN channel was quantized to $K_{\text{ch}}$ levels. The specific quantization levels were found by finely and uniformly quantizing the AWGN channel, then applying the Quantization Algorithm. Then, the message-passing decoder was restricted to using $K$ levels per message ($\log_2 K$ bits per message). For variable nodes, $I = K^{d_v - 1}$, $K_{\text{ch}}$ input messages are quantized into $K$ output messages. For check nodes, $I = K^{d_c - 1}$ input messages are quantized into $K$ output messages.

The results for a $d_c = 6$, $d_v = 3$, rate 1/2 regular LDPC code are given in Fig. 6, which shows noise thresholds versus the degree of channel quantization, for various amounts of message quantization; also shown are unquantized noise thresholds. The most significant observation is that noise thresholds at $K = 16$ levels per message (4 bits per message) are difficult to distinguish from those for which the messages are not quantized. Note also that coarsely quantized messages ($K = 3$) with a finely quantized channel ($K_{\text{ch}} = 12$ to 32) has better noise thresholds than finely quantized messages with a coarsely quantized channel ($K_{\text{ch}} = 2, 3$). This is interesting from a practical point of view, since decoders with coarsely quantized messages have more efficient implementation.

B. Finite-Length LDPC Codes

Fig. 7 shows the bit error-rate for a rate 1/4, $d_v = 3$ and $d_c = 4$ LDPC code of block length 1000, constructed using Gallager’s method [26], on the binary symmetric channel. Again, with $K = 16$ (4 bits per message), the decoder error rate is similar to belief-propagation decoding using a floating point implementation, for smaller BSC crossover probabilities. The decoding maps were generated using a single BSC crossover probability near the noise threshold. But over all the simulated channels, a single sequence of decoding maps was used. Note that error floors, a common problem with quantized decoders, do not appear above probability of error of $10^{-8}$.

C. Algorithm E-like Decoding Mappings

As shown in this subsection, the mappings derived by the proposed method are similar to those of the heuristically-derived Algorithm E [3].

Algorithm E corresponds to the case of $K_{\text{ch}} = 2$ (binary symmetric channel, BSC) and $K = 3$. Algorithm E is a message-passing LDPC decoder for the BSC where the decoder messages set has three messages: 0, 1, and an erasure message denoted by e. The variable node mapping performs
TABLE I: Decoding mappings \( \chi \) and \( \Phi \) produced by the described method, for \( K = 3 \) and \( K_{\text{ch}} = 2 \) (binary symmetric channel), similar to Algorithm E.

(a) Check node mapping \( \chi_1 \)

| message input | output |
|---------------|--------|
| (#0, #e, #1)  | L      |
| (3,0,0)       | 0      |
| (1,0,2)       | 0      |
| (2,1,0)       | e      |
| (1,2,0)       | e      |
| (1,1,1)       | e      |
| (0,3,0)       | e      |
| (0,2,1)       | e      |
| (0,1,2)       | e      |
| (2,0,1)       | l      |
| (0,0,3)       | l      |

(b) Check node mapping \( \chi_2 \)

| message input | output |
|---------------|--------|
| (#0, #e, #1)  | L      |
| (5,0,0)       | 0      |
| (3,0,2)       | 0      |
| (1,0,4)       | 0      |
| (4,0,1)       | 1      |
| (2,0,3)       | 1      |
| (0,0,5)       | 1      |

(c) Variable node mapping \( \Phi_1 \)

| message input | chan. W | output R |
|---------------|---------|----------|
| (#0, #e, #1)  |         |          |
| (2,0,0)       | 0       | 0        |
| (1,1,0)       | 0       | (1,1,0)  |
| (1,0,1)       | 0       | (0,1,0)  |
| (0,2,0)       | 0       | (0,2,0)  |
| (0,1,1)       | 0       | (1,1,0)  |
| (0,0,2)       | 0       | (0,1,0)  |
| (0,0,1)       | 1       | (0,2,0)  |
| (0,0,0)       | 1       | (0,1,1)  |
| (0,0,4)       | 1       | (0,0,5)  |

(d) Variable node mapping \( \Phi_2 \)

| message input | chan. W | output R |
|---------------|---------|----------|
| (#0, #e, #1)  |         |          |
| (2,0,0)       | 0       | 0        |
| (1,1,0)       | 0       | (1,1,0)  |
| (1,0,1)       | 0       | (0,1,0)  |
| (0,2,0)       | 0       | (0,2,0)  |
| (0,1,1)       | 0       | (1,1,0)  |
| (0,0,2)       | 1       | (0,2,0)  |
| (0,0,1)       | 1       | (0,1,1)  |
| (0,0,0)       | 1       | (0,0,5)  |

TABLE II: Per-iteration decoding tables for the BSC with \( K = 3 \) levels per message, for \( d_c = 4 \) and \( d_c = 6 \), tables shown in Table I.

(a) \( d_c = 4 \) and BSC with crossover probability 0.14.

| iteration \( \ell \) | 1 | 2 | 3 | 4 | 5 | 6 |
|----------------------|---|---|---|---|---|---|
| Check node table \( \chi^{(\ell)} \) | \( \chi_1 \) | \( \chi_1 \) | \( \chi_1 \) | \( \chi_1 \) | \( \chi_1 \) | \( \chi_1 \) |
| Var node table \( \Phi^{(\ell)} \) | \( \Phi_1 \) | \( \Phi_1 \) | \( \Phi_2 \) | \( \Phi_1 \) | \( \Phi_2 \) | \( \Phi_2 \) |

(b) \( d_c = 6 \) and BSC with crossover probability 0.06.

| iteration \( \ell \) | 1 | 2 | 3 | 4 | 5 | 6 |
|----------------------|---|---|---|---|---|---|
| Check node table \( \chi^{(\ell)} \) | \( \chi_2 \) | \( \chi_2 \) | \( \chi_2 \) | \( \chi_2 \) | \( \chi_2 \) | \( \chi_2 \) |
| Var node table \( \Phi^{(\ell)} \) | \( \Phi_1 \) | \( \Phi_1 \) | \( \Phi_2 \) | \( \Phi_2 \) | \( \Phi_2 \) | \( \Phi_2 \) |

For concave optimization problems in general, finding efficient algorithms is highly problem-dependent. Indeed, for this specific concave optimization problem, the structure of the mutual information objective function was carefully considered to arrive at an efficient optimization method. Concave optimization results were invoked in the proof of Lemma 2 and related convexity properties are used in the proof of Lemma 4, in the Appendix.

This method applies to arbitrary discrete memoryless channels. Many previous papers concentrated on specific channels, particularly continuous-to-discrete quantization of AWGN channels, and moreover optimal solutions were not obtained. While some DMCs can be obtained by quantizing AWGN channels, arbitrary DMCs which cannot be obtained this way. Thus, our results are in one sense, more general. In addition, the DMC quantization algorithm may be applied to continuous-output channels, such as the binary-input AWGN channel, by first finely quantizing the channel, then applying the Quantization Algorithm to the resulting DMC.

VII. DISCUSSION

This paper showed that finding optimal quantizers for discrete memoryless channels is a concave optimization problem.

Fig. 7: Finite-length simulation on binary symmetric channel. Rate 1/4, block length 1000 (3,6) regular LDPC code.
The optimal quantizer is deterministic, rather than stochastic. As far as we are aware, all previous work assumed deterministic quantizers. From a mathematical viewpoint, there is no reason to *a priori* assume that deterministic quantizers are optimal, and disallow stochastic quantizers; in the rate-distortion problem, the optimal quantizer is usually stochastic. From an engineering viewpoint, deterministic quantizers are far more practical. Lemma 2 shows that this engineering preference is in fact optimal.

This paper’s results are mostly restricted to DMCs with binary inputs. The Quantization Algorithm assumed that the channel outputs can be sorted; but for a channel with three or more inputs, there is no clear sorting of the channel outputs. In fact, the motivation for this paper was optimal quantization of state metrics for the BCJR algorithm (with more than two states) and of non-binary LDPC codes; see earlier work [27] and [28]. These problems could be solved using the quantization of a discrete memoryless channel with non-binary inputs, if such a method existed. Future extensions to non-binary inputs could use techniques similar to those developed here, so this paper forms a basis for solving the non-binary input optimization problem.

Throughout this paper a fixed input distribution was used, which corresponds to finding an achievable rate, for a given channel and a given number of quantization levels. However, computing the capacity of a channel subject to a quantization restriction is of considerable interest. This would require finding an input distribution and quantizer which jointly maximize mutual information. This paper provides a foundation for addressing this problem.

The other major result in this paper was to show that the Quantization Algorithm can be applied to implementation of LDPC decoders. While it has been known for some time that using non-uniform quantizers that change from iteration to iteration can improve performance, the selection of quantizers is ad hoc. A contribution of this paper is to give a systematic approach for generating the quantizers. Indeed, this approach generates quantizers which are non-uniform and change as the iterations progress.

This method selects quantizers and message-passing decoding maps which maximize mutual information at each decoding step. Clearly, it would be more desirable to find decoding maps which maximize mutual information over all iterations. However, as we have shown, this per-iteration optimization has good performance, and yields some insight into efficient quantization methods.

**APPENDIX**

After giving some notation and terminology, this Appendix states and proves Lemma 4. Then, Lemma 5 is stated and proved using Lemma 4. Next, Lemma 3 from Sec. III is proved using Lemma 5. Finally, the Theorem is proved, using Lemma 3.

The theme of Lemmas 3, 4 and 5 is that for any quantizer where the preimage of a quantizer output does not consist of consecutive channel outputs, then there exists another quantizer with higher partial mutual information. Lemma 4 applies for the case of alternately assigning even and odd numbered channel outputs to two quantizer outputs. Lemma 5 applies to the case of any assignment to two quantizer outputs. Finally Lemma 3 applies to an arbitrary number of channel outputs.

A. Preliminaries

Consider some subset of the channel outputs in which all subset members are assigned to one of two quantizer outputs; the size of the subset is \( I' \), with \( I' \leq I \). Let \( c_i = p_{1i} \) and \( b_i = p_{2i} \) for \( i = 1, 2, \ldots, I' \). The channel outputs are indexed and sorted:

\[
\frac{c_1}{b_1} < \frac{c_2}{b_2} < \cdots < \frac{c_{I'}}{b_{I'}}
\]

which corresponds to the condition (11) for the indices \( \{1, 2, \ldots, I'\} \) (see also Remark 1). Let \( C = \sum_{i=1}^{I'} c_i \leq 1 \) and \( B = \sum_{i=1}^{I'} b_i \leq 1 \). For fixed values of \( C \) and \( B \), let \( \mathcal{F}_{C, B} \) be given by:

\[
\mathcal{F}_{C, B} = \{(c, b) \mid 0 \leq c \leq C, 0 \leq b \leq B\},
\]

so that \((c_i, b_i) \in \mathcal{F}_{C, B} \) for \( i = 1, 2, \ldots, I' \).

**Remark 2.** Note that the indexing \( i = 1, 2, \ldots, I' \) is for notational convenience only. In what follows, we could take any subset of size \( I' \) of the \( I \) channel outputs. Clearly if (11) holds for \( I \) channel outputs, then (37) will also hold.

Since this part of the Appendix is concerned with mapping only a subset of channel outputs to two quantizer outputs, we are interested in two assignments \( A_1 \) and \( A_2 \) such that \( A_1 \cup A_2 = \{1, 2, \ldots, I'\} \) (as with Remark 2, the subscripts 1 and 2 should be regarded as a convenience). The terms ‘assignment’ and ‘preimage’ are used interchangeably.

For these two quantizer outputs, define a partial mutual information function \( \iota(c, b) \) for \((c, b) \in \mathcal{F}_{C, B} \) as follows:

\[
\iota(c, b) = c \log \frac{c/p_1}{c+b} + b \log \frac{b/p_2}{c+b} + (C-c) \log \frac{(C-c)/p_1}{(C-c)+(B-b)} + (B-b) \log \frac{(B-b)/p_2}{(C-c)+(B-b)}.
\]

Note that (39) can also be expressed as

\[
\iota(c, b) = c \log \frac{c}{c+b} + b \log \frac{b}{c+b} + (C-c) \log \frac{(C-c)}{(C-c)+(B-b)} + (B-b) \log \frac{(B-b)}{(C-c)+(B-b)} - C \log p_1 - B \log p_2,
\]

where the last two terms on the right-hand side do not depend on the values \( c \) and \( b \).

The partial mutual information function \( \iota(c, b) \) expresses the sum of partial mutual information of two quantizer outputs. By letting \( c = \sum_{i \in A_1} c_i \) and \( b = \sum_{i \in A_1} b_i \) for an assignment \( A_1 \) and \( A_2 \), then it is readily seen that

\[
\iota(c, b) = \iota_1 + \iota_2,
\]

where \( \iota_1 \) and \( \iota_2 \) are the partial mutual information functions for the two subsets.
where \( t_m \) with \( m = 1, 2 \) is defined in (16).

If \( I' = 5 \), \( A_1 = \{1, 2\} \) and \( A_2 = \{3, 4, 5\} \), then the partial mutual information of this assignment is \( \iota(c_1 + c_2, b_1 + b_2) \).

For fixed \( C \) and \( B \) with \( 0 < C \leq 1 \) and \( 0 < B \leq 1 \), the partial mutual information function has the following properties:

(i) For any values \((c, b) \in \mathcal{F}_{C,B}\):
\[
\iota(c, b) = \iota(C - c, B - b). \tag{42}
\]
That is, the function \( \iota(c, b) \) is symmetric with respect to the point \( (C/2, B/2) \).

(ii) For fixed \( p_1 \), the function \( \iota(c, b) \) is convex in \((c, b)\), that is, for any distinct \((c', b') \in \mathcal{F}_{C,B} \) and \((c'', b'') \in \mathcal{F}_{C,B} \),
\[
\iota(c_0, b_0) \leq \theta \iota(c', b') + (1 - \theta) \iota(c'', b'') \tag{43}
\]
for \( 0 \leq \theta \leq 1 \), where \( c_0 = \theta c' + (1 - \theta) c'' \) and \( b_0 = \theta b' + (1 - \theta) b'' \), with equality if and only if
\[
\frac{c'}{b'} = \frac{c''}{b''} = \frac{C}{B}. \tag{44}
\]
Inequality (43) and equation (44) can be shown by the log-sum inequality [4, Theorem 2.7.1]. It follows from the equality condition that the partial mutual information is strictly convex over any sub-region \( \mathcal{F}' \subset \mathcal{F}_{C,B} \) which does not include a segment of the line \( \{ c \} = \frac{c}{b} \), that is, the interior of \( \mathcal{F}' \cap \{ (c, b) \mid \frac{c}{b} = \frac{C}{B} \} \) is empty.

(iii) For any values \((c, b) \in \mathcal{F}_{C,B}\):
\[
\iota(c, b) \geq C \log \frac{C}{C + B} + B \log \frac{B}{C + B} - C \log p_1 - B \log p_2, \tag{45}
\]
which holds with equality if and only if \( \frac{c}{b} = \frac{C}{B} \).

Equation (45) can be easily shown by applying the log-sum inequality to the first and the third terms and the second and the forth terms on the right-hand side of (40), respectively.

Define the following points,
\[
q_i = (c_i, b_i), \quad \text{for} \quad i = 1, \ldots, I', \tag{46}
\]
so that \( q_i \in \mathcal{F}_{C,B} \). These points and the region \( \mathcal{F}_{C,B} \) are illustrated in Fig. 8 and Fig. 9, for \( I' = 4 \). If the quantizer consists of \( A_1 = \{i\} \) and \( A_2 = \{1, 2, 3, 4\} \setminus \{i\} \), then the partial mutual information for this assignment can be seen in the graph as the value of the level curves at \( q_i = (c_i, b_i) \). Similarly, if \( A_1 = \{1, 3\} \) and \( A_2 = \{2, 4\} \), then the partial mutual information can be seen as the value of the level curves at \( q_{13} = (c_1 + c_3, b_1 + b_3) \).

Fig. 8 and Fig. 9 illustrate examples for two important cases. Fig.8 shows the case of \( \frac{b_1 + b_3}{c_1 + c_3} > \frac{b_2}{C} \), and Fig. 9 shows the case of \( \frac{b_1 + b_3}{c_1 + c_3} < \frac{b_2}{C} \). The sorting assumption (37) implies that the slope of the line passing through \((c_i, b_i)\) and \((0, 0)\) decreases as \( i \) increases. The diagonal line corresponds to the points satisfying \( \frac{c}{b} = \frac{b}{C} \), and partial mutual information on this line is the minimum in \( \mathcal{F}_{C,B} \), by property (iii).

Now, define the following terminology. An “odd-even assignment” is an assignment where the \( I' \) channel outputs indexed \( \{1, 2, 3, \ldots, I'\} \) are alternately assigned to two quantizer outputs 1 and 2 with sets \( A_1 \) and \( A_2 \). The odd-even assignment is \( A_1 = \{1, 3, \ldots, I'\} \) and \( A_2 = \{2, 4, \ldots, I' - 1\} \) (for \( I' \) odd), or \( A_1 = \{1, 3, \ldots, I - 1\} \) and \( A_2 = \{2, 4, \ldots, I'\} \) (for \( I' \) even).

In an arbitrary mapping to two quantizer outputs, a consecutive sequence of channel outputs may be mapped to the same quantizer output. Any maximal such set is called a “output group.” Denote by \( H \) the number of such groups. For example, if eight channel outputs are assigned
\[
A_1 = \{1, 2, 3, 6, 7\} \quad \text{and} \quad A_2 = \{4, 5, 8\}, \tag{47}
\]
then \( H = 4 \). The four output groups are \( \{1, 2, 3\} \), \( \{4, 5\} \), \( \{6, 7\} \), and \( \{8\} \). The minimum value of \( H \) is two. For the odd-even assignment, \( H = I' \).

An assignment \( A_1 \) and \( A_2 \) is called “consecutive” if \( \max(A_1) < \min(A_2) \). Otherwise, the assignment is called “non-consecutive.” An assignment is consecutive if and only if the number of output groups \( H = 2 \). For example, the assignment \( A_1 = \{1, 3\} \) and \( A_2 = \{4, 5, 7\} \) is consecutive. On the other hand, \( A_1 = \{1, 4\} \) and \( A_2 = \{3, 5, 7\} \) is a non-consecutive assignment.

B. Lemma 4

The following lemma states that reducing the number of output groups will increase partial mutual information.

**Lemma 4.** For the odd-even assignment with \( I' \geq 3 \) channel outputs satisfying (37) and \( H' = I' \) output groups, let the partial mutual information be denoted by \( \iota' \). Then, there exists another assignment with \( H'' \) output groups and partial mutual information \( \iota'' \), with \( H'' < H' \) and \( \iota'' > \iota' \).

To illustrate Lemma 4, consider \( I' = 3 \). There are three non-trivial ways to assign the three channel outputs to two quantizer outputs:

- **Quantizer 1:** combine outputs 2 and 3 (\( H = 2 \) output groups),
- **Quantizer 2:** combine outputs 1 and 3 (odd-even; \( H = 3 \) output groups),
- **Quantizer 3:** combine outputs 1 and 2 (\( H = 2 \) output groups).

Partial mutual information for each quantizer is \( \iota(c_1, b_1) \), \( \iota(c_2, b_2) \), and \( \iota(c_3, b_3) \), respectively. Where property (i) was used for Quantizer 2. From Lemma 4, Quantizer 1 or Quantizer 3 always has greater partial mutual information than Quantizer 2. That is,
\[
\iota(c_2, b_2) < \max \{ \iota(c_1, b_1), \iota(c_3, b_3) \}. \tag{48}
\]
Quantizer 1 and Quantizer 3 each have two output groups, which is strictly fewer than the three output groups of Quantizer 2.

Intuitively, it is expected that the partial mutual information for some channel with \( H' = 4 \) output groups should be lower than the same channel with \( H' = 3 \) output groups. But for mathematical completeness, Lemma 4 is proved for arbitrary \( H' \). The proof for \( I' = 3 \) has some simplifications. So it will be proved for \( I' = 4 \) first, then extended to higher values.
Proof. Consider $I' = 4$ and $H' = 4$. The odd-even assignment has partial mutual information $\iota_{13} = \iota(c_1 + c_3, b_1 + b_3)$. Consider cases of three other assignments:

- Quantizer 1: $A_1 = \{1, 2, 3\}$ and $A_2 = \{4\}$, has $H'' = 2$, with partial mutual information $\iota_{123} = \iota(c_1 + c_2 + c_3, b_1 + b_2 + b_3)$,
- Quantizer 2: $A_1 = \{1\}$ and $A_2 = \{2, 3, 4\}$, has $H'' = 2$, with partial mutual information $\iota_1 = \iota(c_1, b_1)$, and
- Quantizer 3: $A_1 = \{1, 2\}$ and $A_2 = \{3, 4\}$, has $H'' = 2$, with partial mutual information $\iota_{12} = \iota(c_1 + c_2, b_1 + b_2)$.

It will be shown that the partial mutual information resulting from one of these three assignments has greater mutual information than the odd-even assignment, that is:

$$\iota_{13} < \max(\iota_1, \iota_{123}, \iota_{12}).$$

Since for each these three cases, $H'' < H'$, then the Lemma will hold for $I' = 4$.

Define the points $q_{13}, q_{123} \in C, B$ as:

$$q_{13} = (c_1 + c_3, b_1 + b_3) \text{ and } q_{123} = (c_1 + c_2 + c_3, b_1 + b_2 + b_3).$$

Define the following rays:

$$R_0 = \{\theta q_{13} | \theta \geq 1\}, \quad R_1 = \{q_{13} + \theta (q_{123} - q_{13}) | \theta > 0\} \text{ and } R_2 = \{q_{13} + \theta (q_{123} - q_{13}) | \theta \leq 0\}.$$

The union of $R_1$ and $R_2$ is a line, which has a minimum in partial mutual information. The minimum is either in $R_1$ or in $R_2$. We consider the following three cases, corresponding to the three quantizers:

- **Case 1** The minimum is in $R_2$. Then by (strict) convexity of partial mutual information, property (ii):

$$\iota_{13} < \iota_{123}.$$  \hfill (55)

- **Case 2** The minimum is in $R_1$, and $\frac{b_1 + b_3}{c_1 + c_3} > \frac{b_2}{c_2}$; please refer to Fig. 8. Then we can show:

$$\iota_{13} < \iota_1,$$  \hfill (56)

as follows. Since the minimum in $R_1 \cup R_2$ is in $R_1$, then the minimum in $R_2$ alone is at $q_{13}$. The line coincident with $R_0$ passes through the origin, and the origin is the minimum along this line, by property (iii). So, by similar convexity arguments, the minimum along $R_0$ is also at $q_{13}$. In other words, all the points on $R_2 \setminus q_{13}$ and on $R_0 \setminus q_{13}$ have greater partial mutual information than $q_{13}$.

The two rays $R_0$ and $R_2$ define a cone $D$ with vertex $q_{13}$. Observe that if $\iota(q_{13}) \leq \iota(z_i)$, for all $z_i \in R_i$ with $i = 0, 2$, then $\iota(q_{13}) \leq \iota(x)$ for all $x \in D$. That is, the partial mutual information for any point inside $D$ is greater than that for the vertex $q_{13}$. This claim can be verified as follows: Consider a ray $R_3$ that is defined as

$$R_3 = \{(c, b) | c = c_1 + c_3, b \geq b_1 + b_3\}.$$  \hfill (57)
Then the cone $D$ is divided into two cones with the vertex $q_{13}$: the cone $D_0$ defined by the rays $R_0$ and $R_3$ and the cone $D_2$ defined by the rays $R_2$ and $R_4$. For any point $q' \in D_0$, the line, denoted by $l_1(q')$, passes through $q'$ and $q_{13}$ crosses the line $\frac{b}{c} = \frac{p}{r}$. This means that all points in the intersection of $l_0(q')$ and $D_0$ has larger partial mutual information than that for $q_{13}$, and thus, the partial mutual information for any point in $D_0 \setminus \{q_{13}\}$ is greater than that for the vertex $q_{13}$. Now, we turn to considering $D_2$. For any point $q' \in D_2$, the vertical line, denoted by $l_2(q')$, passing through $q'$ also crosses the line $\frac{b}{c} = \frac{p}{r}$. Let $z_2$ denote the intersection of $l_2(q')$ and $R_0$. Then from the same reasoning, the partial mutual information for the point $q' \in D_2$ is greater than that for $z_2$, which is greater than that for $q_{13}$, since $z_2$ is on $R_0$.

All that remains is to show that $q_1$ is inside the cone $D$. We use a geometrical argument using Fig. 8, for fixed values of $C$ and $B$. The points $q_i$ for $i \in \{1, \ldots, 4\}$, $q_{13}$, and $q_{123}$ are plotted. In addition, the cone formed by $R_0$ and $R_2$ is shown. Consider a ray $R'$ defined as

$$R' = \{ q_1 + \theta(q_{13} - q_1) | \theta > 0 \},$$

which stems from $q_1$ and passes through $q_{13}$. Geometrically, it is clear that if the slope of $R'$ is less than the slope of $R_2$, then $q_1$ is inside this cone. The slope of $R'$ is $\frac{b}{c}$ and the slope of $R_2$ is $\frac{b_2}{c_2}$. This holds by assumption (37). Thus, $q_1$ is inside the cone and $\iota_{13} < \iota_1$.

Case 3 The minimum is in $R_1$, and $\frac{b_1 + b_2}{c_1 + c_3} < \frac{b_2}{c_2}$; please refer to Fig. 9. In this case, $q_{13}$ is always below the line $\frac{b}{c} = \frac{p}{r}$, since otherwise the ray $R_2$ crosses the line $\frac{b}{c} = \frac{p}{r}$ and the point having the minimum partial mutual information is always in $R_2$. Using the same reasoning as in Case 2, it can be shown that $q_3$, rather than $q_1$, is inside the cone defined by $R_0$ and $R_2$ (see Fig. 9). Further, the point $q_{34} = q_3 + q_4$ is also inside this cone since the slope of $q_4$ is less than that of $R_2$. Property (i) indicates $\iota_{34} = \iota(c_3 + c_4, b_3 + b_4)$, and we can show

$$\iota_{13} < \iota_{34}. \quad (59)$$

Since at least one of Case 1, Case 2 or Case 3 always holds, then (49) always holds for $I' = 4$.

Finally, the above argument is extended to arbitrary $I'$. The partial mutual information of the odd-even assignment, which has $H'' = I'$, is denoted by $\iota_0$. The generalization of the three cases are:

- **Quantizer 1**: $\mathcal{A}_1 = \{1, 2, 3, 5, 7, \ldots\}$ and $\mathcal{A}_2 = \{4, 6, 8, \ldots\}$, has $H'' = H' - 2$, with partial mutual information $\iota_0$:
  $$\iota_0 = \iota(\sum_{i \in \mathcal{A}_1} c_i, \sum_{i \in \mathcal{A}_2} b_i). \quad (60)$$

- **Quantizer 2**: $\mathcal{A}_1 = \{1\}$ and $\mathcal{A}_2 = \{2, 3, 4, 5, \ldots\}$, has

![Fig. 9: Partial mutual information $\iota(c, b)$ is shown using level curves. Case 3](image-url)
The assignments can continue until the number of output groups is two, corresponding to a consecutive assignment. Lemma 4 guarantees that, at each step of the recursion, the partial mutual information increases. This completes the proof of Lemma 5.

D. Proof of Lemma 3

To prove Lemma 3, it is sufficient to show that for any quantizer which contains at least one non-consecutive assignment, there exists another consecutive quantizer, which has greater mutual information. Define “pairwise consecutive” as any two output groups which satisfy the definition of consecutive given earlier.

Let $Q, Q', Q'', \ldots$ be a sequence of quantizers that respectively produce total mutual information $i, i', i'', \ldots$. Assume that $Q$ has at least one pair of output groups which are not pairwise consecutive. Select one such pair. By Lemma 5, there exists a new quantizer $Q'$, obtained by relabeling only that pair, which has strictly greater mutual information, $i' > i$. Note that $Q'$ may still have pairs of output groups which are not pairwise consecutive. Lemma 5 can be applied recursively, obtaining a sequence of quantizers $Q, Q', Q'', \ldots, Q^{\text{end}}$ which have total mutual information satisfying $i < i' < i'' < \cdots < i^{\text{end}}$.

The recursion terminates with the quantizer $Q^{\text{end}}$, which is the first quantizer which has all assignments pair-wise consecutive. Note that the recursion is guaranteed to terminate because there are a finite number of possible quantizers. Furthermore, since Lemma 5 provides a strict inequality, the new mutual information is higher than all previous values, and thus the new quantizer is distinct. That is, a quantizer can appear in the sequence only one time. Thus, a quantizer with non-consecutive labeling cannot maximize mutual information at the quantizer output. This completes the proof of Lemma 3.

E. Proof of Theorem’s Optimality Part

In this section, the optimality part of the Theorem is proved. Lemma 3 is a necessary condition on quantizer optimality. In particular, the optimal quantizer satisfies the condition that $a_1 < a_2 < \cdots < a_{K-1}$, where $a_i$ denote the output group boundaries. This is the exact condition over which the Quantization Algorithm searches. Thus, it is sufficient to show that among these quantizers, the Quantization Algorithm, which is the realization of a dynamic program, will output the quantizer which maximizes mutual information.

In the language of dynamic programming, a problem exhibits optimal substructure if the optimal solution contains optimal solutions to subproblems. If this condition holds, then dynamic programming provides the optimal solution, and moreover, the optimal substructure should be exploited in the optimization [29, Sec. 15.3].

For the Theorem, the subproblem consists of finding the quantizer which maximizes partial mutual information for some partial quantization of the outputs. In detail, recall $S_k(a)$ is the maximum of partial mutual information when channel outputs $1$ to $a$ are quantized to quantizer outputs $1$ to $k$,

$$S_k(a) = \max_{a'} \left( S_{k-1}(a') + \iota(a' \rightarrow a) \right).$$

Thus, the recursion can be used to efficiently compute the optimal quantizer. This completes the proof of the optimality part of the Theorem.
where the maximization is over $a' \in \{k-1, \ldots, a-1\}$.

For fixed $k$ and $a$, assume that $S_k(a)$ is the maximum of partial mutual information, corresponding to the optimal quantization of channel outputs $1$ to $a$ to the quantizer output groups $1$ to $k$. Let $\bar{a}$ be the last element of group $k-1$, that is:

$$S_k(a) = S_{k-1}(\bar{a}) + \iota(\bar{a} \to a),$$  \hspace{1cm} (65)

so that group $k$ consists of $\bar{a} + 1, \ldots, a$, that is, $A_k = \{\bar{a} + 1, \ldots, a\}$. Then, the quantizer for channel outputs $1$ to $\bar{a}$ must also be optimal. This is true because if another quantizer of $1$ to $\bar{a}$ produced higher mutual information, then the quantization of $1$ to $a$ would also have higher partial mutual information, leading to a contradiction of the assumption that $1$ to $a$ was optimally quantized.

The above argument is sufficient to prove that the quantization problem has optimal substructure, and since the Quantization Algorithm exploits this structure, the algorithm is optimal. Along with the earlier statement that the complexity is proportional to $I^2$, the proof of the Theorem is completed.

REFERENCES

[1] B. Kurkoski and H. Yagi, “Concatenation of a discrete memoryless channel and a quantizer,” in *Proceedings of the IEEE Information Theory Workshop*, (Cairo, Egypt), pp. 160–164, January 2010.
[2] B. Kurkoski, K. Yamaguchi, and K. Kobayashi, “Noise thresholds for discrete LDPC decoding mappings,” in *Proceedings IEEE Global Telecommunications Conference*, (New Orleans, USA), pp. 1–5, November–December 2008.
[3] T. J. Richardson and R. L. Urbanke, “The capacity of low-density parity check codes under message-passing decoding,” in *IEEE Transactions on Information Theory*, vol. 47, pp. 599–618, February 2001.
[4] T. M. Cover and J. A. Thomas, *Elements of Information Theory*. Wiley, 1991.
[5] S. Arimoto, “An algorithm for computing the capacity of arbitrary discrete memoryless channels,” in *IEEE Transactions on Information Theory*, vol. 18, pp. 14–20, January 1972.
[6] R. E. Blahut, “Computation of channel capacity and rate-distortion functions,” in *IEEE Transactions on Information Theory*, vol. 18, pp. 460–473, July 1972.
[7] N. Tishby, F. C. Pereira, and W. Bialek, “The information bottleneck method,” in *Proceedings 37th Annual Allerton Conference on Communication, Control, and Computing*, (Monticello, IL, USA), September 1999.
[8] J. M. Wozencraft and R. S. Kennedy, “Modulation and demodulation for probabilistic coding,” in *IEEE Transactions on Information Theory*, vol. 12, pp. 291–297, July 1966.
[9] J. M. Wozencraft and I. M. Jacobs, *Principles of Communication Engineering*. Prospect Heights, IL, USA: Waveland Press Inc., 1990. Originally Published 1965 by Wiley.
[10] J. L. Massey, “Coding and modulation in digital communications,” in *Proceedings of International Zurich Seminar on Digital Communications*, (Zurich, Switzerland), pp. E2(1)–E2(4), 1974.
[11] L. Lee, “On optimal soft-decision demodulation,” in *IEEE Transactions on Information Theory*, vol. 22, pp. 437–444, July 1976.
[12] J. Salz and E. Zehavi, “Decoding under integer metrics constraints,” in *IEEE Transactions on Communications*, vol. 43, pp. 307–317, February/March/April 1995.
[13] G. Jeong and D. Hsia, “Optimal quantization for soft-decision turbo decoding,” in *Proceedings IEEE Vehicular Technology Conference*, pp. 1620–1624, September 1999.
[14] X. Ma, X. Zhang, H. Yu, and A. Kavcic, “Optimal quantization for soft-decision decoding revisited,” in *International Symposium on Information Theory and its Applications*, (Xi’an, China), October 2002.
[15] A. D. Liveris and C. N. Georgiades, “On quantization of low-density parity-check coded channel measurements,” in *Proceedings IEEE Global Telecommunications Conference*, (San Francisco, CA, USA), pp. 1649–1653, December 2003.
[16] J. Singh, O. Dabeer, and U. Madhow, “On the limits of communication with low-precision analog-to-digital conversion at the receiver,” in *IEEE Transactions on Communications*, vol. 57, pp. 3629–3639, December 2009.
[17] Z. Zhang, L. Dolecek, B. Nikolic, V. Anantharam, and M. Wainwright, “Design of LDPC decoders for improved low error rate performance: quantization and algorithm choices,” in *IEEE Transactions on Communications*, vol. 57, pp. 3258–3268, November 2009.
[18] X. Wu, C. Ling, M. Jiang, E. Xu, C. Zhao, and X. You, “New insights into weighted bit-flipping decoding,” in *IEEE Transactions on Communications*, vol. 57, pp. 2177–2180, August 2009.
[19] D. V. Nguyen, B. V. Vasic, and M. W. Marcellin, “Two-bit bit flipping decoding of LDPC codes.” Submitted to ISIT 2011. Available http://arxiv.org/abs/1103.0801.
[20] J.-S. Lee and J. Thorpe, “Memory-efficient decoding of LDPC codes,” in *Proceedings of IEEE International Symposium on Information Theory*, (Adelaide, Australia), pp. 459–463, September 2005.
[21] R. Horst and P. M. Pardalos, eds., *Handbook of Global Optimization*. Kluwer Academic Publishers, 1995.
[22] R. Horst, P. M. Pardalos, and N. V. Thoai, *Introduction to Global Optimization*. Kluwer Academic Publishers, 1995.
[23] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge University Press, 2004.
[24] Source code for quantization of discrete memoryless channels. Online http://www.lit.ice.ucc.ac.jp/kurkoski/quantizer.html.
[25] C.-C. Wang, S. R. Kulkarni, and H. V. Poor, “Density evolution for asymmetric memoryless channels,” in *IEEE Transactions on Information Theory*, vol. 51, pp. 4216–4236, December 2005.
[26] R. G. Gallager, *Low-Density Parity-Check Codes*. Cambridge, MA, USA: The M.I.T. Press, 1963.
[27] B. Kurkoski, K. Yamaguchi, and K. Kobayashi, “On BCJR state metric quantization for turbo equalization,” in *Proceedings of IEEE International Symposium on Information Theory*, (Adelaide, Australia), June 2005.
[28] B. M. Kurkoski, K. Yamaguchi, and K. Kobayashi, “Density evolution for GF(q) LDPC codes via simplified message-passing sets,” in *Information Theory and Applications Workshop, 2007* (La Jolla, CA, USA), pp. 237–244, January 2007.
[29] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to Algorithms, Second Edition*. The MIT Press, 2001.