Aalborg Universitet

Decoding of LDPC codes with binary vector messages and scalable complexity

Lechner, Gottfried; Land, Ingmar; Rasmussen, Lars

Published in:
5th International Symposium on Turbo Codes and Related Topics, 2008

DOI (link to publication from Publisher):
10.1109/TURBOCODING.2008.4658724

Publication date:
2008

Document Version
Publisher's PDF, also known as Version of record

Link to publication from Aalborg University

Citation for published version (APA):
Lechner, G., Land, I., & Rasmussen, L. (2008). Decoding of LDPC codes with binary vector messages and scalable complexity. In *5th International Symposium on Turbo Codes and Related Topics, 2008* (pp. 350-355). IEEE (Institute of Electrical and Electronics Engineers). https://doi.org/10.1109/TURBOCODING.2008.4658724

General rights
Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal -

Take down policy
If you believe that this document breaches copyright please contact us at vbn@aub.aau.dk providing details, and we will remove access to the work immediately and investigate your claim.

Downloaded from vbn.aau.dk on: December 04, 2025

Decoding of LDPC Codes with Binary Vector Messages and Scalable Complexity

Gottfried Lechner*, Ingmar Land†‡, Lars Rasmussen†

* Telecommunications Research Center Vienna (ftw.), Vienna, Austria
Email: {lechner@ftw.at}
† Institute for Telecommunications Research, University of South Australia, Adelaide, Australia
Email: {ingmar.land,lars.rasmussen}@unisa.edu.au
‡ Department of Electronic Systems, Aalborg University, Aalborg, Denmark

***Abstract*—In this paper, an iterative decoder for LDPC codes, suitable for high-speed processing, is proposed and analyzed. The messages exchanged between the nodes are binary vectors, where the information is conveyed by the number of ones rather than their positions in the vector. Regarding this aspect, the approach exhibits some similarities to stochastic decoding techniques. The check node decoders perform simple bit-wise modulo-2 additions, whereas the variable node decoders perform more complex processing tasks, making the resulting structure attractive for high-speed hardware implementation. By selecting the length of the binary vector messages between one and infinity, the complexity and the performance of the decoder can be scaled between that of the original binary message passing algorithm and that of the sum-product algorithm. Density evolution is developed for the proposed decoding algorithm, and decoding thresholds are determined as functions of the length of the binary vector messages. Simulation results exemplify the performance for finite-length codes.**

I. Introduction

Low-density parity-check (LDPC) codes and their iterative decoding algorithms were introduced in [1], [2]. LDPC codes have since been shown to perform close to channel capacity when decoded with the sum-product algorithm (SPA), in which the messages are probabilities or L-values [3]. Efficient code design tools have been developed based on extrinsic information transfer (EXIT) charts [4] or density evolution [5], [6], allowing for code structures performing increasingly closer to capacity. For applications operating at very high transmission rates, however, the application of the SPA may be prohibitive complex. As an alternative binary message passing algorithms have been considered.

In [7], Gallager's decoding algorithm B is generalized by allowing for the variable node decoder to perform its computations in the L-value domain, while still restricting the messages between variable and check nodes to be binary. With this generalization it is possible to combine binary messages from the check nodes with soft decisions from the channel, thereby improving the performance of the decoding algorithm if soft decisions from the channel are available. This method is referred to as the binary message passing (BMP) algorithm in the following.

In the present paper, the approach in [7] is further developed. We still restrict the messages between variable nodes and check nodes to be binary; however, rather than sending only a single hard decision, we allow each check node decoder to send a sequence of binary messages before the variable node decoders are activated. The message sent by a variable node is represented as a (randomly permuted) binary sequence where the fraction of zeros and ones correspond to the quantized probability of the code digit being a zero or one, respectively. Since a check node decoder receives sequences of binary messages, it simply computes the bitwise modulo-2 addition of all incoming messages, allowing for an efficient small-sized and high-speed implementation. The check nodes sequentially forward the resulting updated binary messages. Each variable node decoder therefore receives sequences of hard decisions from the check nodes, which are subsequently combined into L-values as described below. As we will show, this algorithm corresponds to BMP [7] if the length of the sequences is one and to optimal SPA decoding if the length of the sequences tends to infinity. Therefore, the complexity and performance of the resulting iterative decoder can easily be scaled by selecting the length of the sequences.

Although this approach is motivated by efficient implementation of high-speed binary check node decoders, we will represent all sequences as vectors and assume that the check node decoder receives the entire vector instantly and performs bit-wise operations only. This equivalent representation is more convenient for the analysis and the corresponding algorithm is referred to as *binary vector message passing (BVMP)* in the following.

Since we use a random permutation of the elements of a message, the decoder is no longer deterministic and has some relation to stochastic decoding [8], [9]. In stochastic decoding the input to the iterative decoder consists of binary messages that are (randomly) generated by binary white sources having distributions according to the channel L-values; the check node decoders perform a binary addition, which is a deterministic operation; the output messages of the variable node decoders are determined in a deterministic or stochastic way, depending on the configuration of the input messages. For details, we

refer the reader to [8], [9] and the references therein.

In [8], packetized super-nodes are proposed to avoid that parts of the decoding graph get stuck in a certain state, called latching, which prevents further improvement of the decoding output over iterations. In so-called packetized super-nodes, the distribution of the stochastically generated messages depends on the incoming messages within a certain time-frame. This concept is obviously related to the variable node decoders in our approach; however, we use the message length to scale complexity and performance, which is based on fundamentally different principles as compared to [8].

The main contribution of our work is to derive a theoretical framework for such algorithms, avoiding the use of heuristic scaling factors which are essential for the approach in [9]. The analytical framework allows for the use density evolution to optimize the operation of the variable node decoders with respect to the iteration number. Density evolution is also used to determine the decoding threshold as a function of the message length, which allows to determine analytically the trade-off between complexity and performance.

The remainder of the paper is structured as follows. In Section II, the system model is introduced, while the proposed decoder with binary vector messages is described in Section III. In Section IV, we derive density evolution for the proposed decoder, and corresponding decoding thresholds and simulation results for finite-length codes are presented in Section V. Concluding remarks summarize the paper in Section VI.

Throughout the paper, random variables are denoted by uppercase letters and their realizations are denoted by lowercase letters.

II. System Model

The following system model is employed. Consider a regular LDPC code of length N. The variable node degree is denoted by d_v, and the check node degree is denoted by d_c. A generalization to irregular LDPC codes is straight-forward.

The codewords are assumed to be uniformly distributed, and transmitted over a symmetric memoryless communication channel, e.g., an AWGN channel. The code bits are denoted by X_n and the corresponding channel outputs are denoted by y_n, $n = 1, \ldots, N$. For each y_n, the channel L-value

$$l_{\mathsf{ch},n} := L(X_n|y_n) = \ln \frac{\Pr(X=0|y_n)}{\Pr(X=1|y_n)}$$

is computed. These channel L-values are given to the iterative decoder. In the following discussions, the index n may be dropped for convenient notation.

III. BVMP Decoder

This section describes the proposed new binary vector message passing (BVMP) algorithm. The variable node decoders and the check node decoders exchange messages that are binary vectors of length Q. The fraction of zeros in a vector represents the probability for a code bit being a zero, and the fraction of ones represents the probability of a code bit being a one. Thus a binary vector message can be interpreted as a representation of a quantized probability, where only the number of ones matters but not their positions within the vector, i.e., only the Hamming weights of the messages are relevant.

The variable nodes and the check nodes exchange messages until a certain maximum number of iterations is reached or the decoding result is a codeword. For convenient notation, we explain the decoding operations only for an individual variable node and for an individual check node.

A. Variable Node Decoder

Consider a variable node of degree d_v. The associated code bit is denoted by X. Further, denote an extrinsic message as $\boldsymbol{b}_{ev} \in \{0,1\}^Q$, the $(d_v - 1)$ incoming messages (from the other check nodes) as $\boldsymbol{b}_{av,j} \in \{0,1\}^Q$, $j = 1, \ldots, d_v - 1$, and the channel L-value as l_{ch}. The extrinsic message is computed in three steps:

(i) convert all messages to L-values;
(ii) perform the variable node operation in the L-value domain; and
(iii) convert all messages back to binary vectors.

In the first step, the incoming binary vector messages are converted into L-values $L(X|\boldsymbol{b}_{av,j})$. Let $w_{av,j} = w_H(\boldsymbol{b}_{av,j})$, $j = 1, \ldots, d_v - 1$, denote the Hamming weights, i.e., the number of ones, in the binary vectors. Since only the Hamming weights of the vectors matters, we have

$$\begin{aligned} l_{av,j} := L(X|\boldsymbol{b}_{av,j}) &= L(X|w_{av,j}) \\ &= \ln \frac{p(w_{av,j}|X=0)}{p(w_{av,j}|X=1)} \\ &= \ln \frac{p(w_{av,j}|X=0)}{p(Q-w_{av,j}|X=0)}. \end{aligned} \tag{1}$$

In the last line we use the symmetry

$$p(w|X=1) = p(Q-w|X=0), \tag{2}$$

which results directly from the symmetry of the conditional distribution of the L-values due to the symmetry of the communication channel. To compute these L-values, the conditional distributions $p(w_{av,j}|X = 0)$ have to be known. We will determine these distributions by density evolution in Section IV. Notice that by doing so, density evolution is used to determine the optimal conversion from Hamming weights of the binary vector messages to L-values, and thus for decoder optimization.

In the second step, the L-values are added to obtain the extrinsic L-value

$$l_{ev} = l_{\mathsf{ch}} + \sum_{j=1}^{d_v-1} l_{av,j}, \tag{3}$$

similar to the operation of the optimal message passing algorithm.

In the third step, the extrinsic L-value is converted into a binary vector. First, the probability for $X = 1$ is computed from the L-value:

$$p_{ev} := \Pr(X = 1|l_{ev}) = \frac{1}{1 + e^{l_{ev}}}. \tag{4}$$

From this, the number of ones, w_{ev}, in the binary vector $\boldsymbol{b}_{ev}$ is determined as

$$w_{ev} := \mathrm{round}(p_{ev}Q), \tag{5}$$

where the function $\mathrm{round}(\cdot)$ denotes rounding to integers, i.e., for all integers a, $\mathrm{round}(a') = a$ if $a' \in [a - 0.5, a + 0.5)$. Notice that w_{ev}/Q corresponds to the quantized value of p_{ev}. The vector $\boldsymbol{b}_{ev}$ is then obtained by randomly permuting a vector with w_{ev} ones and $Q - w_{ev}$ zeros,

$$\boldsymbol{b}_{ev} := \mathrm{perm}([\underbrace{1\ldots1}_{w_{ev}}\underbrace{0\ldots0}_{Q-w_{ev}}]), \tag{6}$$

where $\mathrm{perm}(\cdot)$ denotes a random permutation. Possible implementation of the required random permutations is discussed in [9], [10].

Notice that only the positions of the ones within the vector is random, whereas the number of ones (the Hamming weight) is deterministic. This makes sure that the BMP algorithm is obtained for message length $Q = 1$.

B. Check Node Decoder

Consider a check node of degree d_c. Denote an extrinsic message as $\boldsymbol{b}_{ec}$, and the $(d_c - 1)$ incoming messages (from the other variable nodes) as $\boldsymbol{b}_{ac,j}$, $j = 1, \ldots, d_c - 1$. The check node decoder performs a bit-wise modulo-2 addition, i.e.,

$$\boldsymbol{b}_{ec} = \bigoplus_{j=1}^{d_c-1} \boldsymbol{b}_{ac,j}. \tag{7}$$

Since we restricted the check node decoder to bit-wise operations, this is the optimal processing rule [11].

Notice that a more complicated check node operation is necessary if the binary vector messages were quantized L-values. Only due to the application of random permutations, the simple bit-wise operation is optimal.

C. Binary Message Passing and SPA

Due to the definition of the BVMP decoding algorithm, it is equivalent to the BMP algorithm for $Q = 1$ and to the SPA for $Q \to \infty$.

For the case $Q = 1$, the random permutation in Eqn. (6) can be removed and the computation of the L-value in Eqn. (1) corresponds to the L-value computation for a binary symmetric channel. This leads to the BMP decoding algorithm [7].

For the case $Q \to \infty$, the weights represent the exact probabilities without quantization errors, and this leads to optimal processing at the variable nodes. The check node operation is also optimal for this case. To see this, assume first a check node of degree three where the fraction of ones at the two inputs is denoted as p and q, respectively. The fraction of ones at the output is computed as $p(1-q)+q(1-p)$ which is the sum-product update rule [3]. Thus in this case, the BVMP algorithm corresponds to the SPA for $Q \to \infty$. The same holds for check nodes of higher degrees, since these can be recursively formed from degree three check nodes.

IV. Density Evolution

All information exchanged between the variable-node decoder and the check-node decoder is represented by the weights of the binary vector messages exchanged. Therefore, in order to determine the performance and the convergence behavior of the decoder, we use density evolution to track the probability mass function of the message weights. Since we are dealing with discrete distributions (for finite Q), we can use discretized density evolution without loss of accuracy.

The density transfer functions of the variable node decoder and the check node decoder are developed below. The results are then used to determine the decoding thresholds as functions of the message lengths Q, as well as for the implementation of the actual decoder as explained in Section III-A. In the derivations below, the following indices notation is used: "w" for Hamming weight, "l" for L-value, "a" for a-priori, "e" for extrinsic, "v" for variable node, and "c" for check node.

A. Variable Node Decoder

Let p_{wav} denote the conditional probability mass function of the weights of the messages at the input of a variable node. Using (1), we compute the probability density function p_{lav} of the corresponding L-values.

$$p_{lav}(l) = \sum_{k=0}^{Q} \delta\left(l - \ln\frac{p_{wav}(k)}{p_{wav}(Q-k)}\right) p_{wav}(k), \tag{8}$$

where $\delta(\cdot)$ evaluates to one if its argument is zero and to zero elsewhere.

The density p_{lev} of the extrinsic L-values is obtained by the convolution of all incoming densities from the check nodes and the density of the L-values from the channel denoted by p_{ch}:

$$p_{lev}(l) = p_{lav,1} * \ldots * p_{lav,d_v-1} * p_{ch}, \tag{9}$$

where $*$ denotes convolution.

The probability mass function of the weights of the extrinsic messages can be computed in the following way. Due to (5), we obtain $w_{ev} = w$ if $p_{ev} \in [\frac{w}{Q+1}, \frac{w+1}{Q+1})$, $w = 0, 1, \ldots, Q$. Denote the bounds by $\rho_w = \frac{w}{Q+1}$ and their L-value equivalents by[1] $\zeta_w = \ln\frac{\rho_w}{1-\rho_w} = \ln\frac{w}{Q+1-w}$ for $w = 0, \ldots, Q+1$. Then we obtain the desired probabilities by

$$p_{wev}(w) = \int_{\zeta_w}^{\zeta_{w+1}} p_{lev}(l)dl, \tag{10}$$

as $w_{ev} = w$ if $l_{ev} \in [\zeta_w, \zeta_{w+1})$.

[1]We define $\ln 0 := -\infty$ for convenience.

B. Check Node Decoder

Consider first a degree three check node. We have two incoming messages with weights w_{ac1} and w_{ac2}, respectively. Without loss of generality we assume $w_{ac1} \geq w_{ac2}$. The conditional probability that the extrinsic message of a check node has weight w_{ec} is given by

$$p(w_{ec}|w_{ac1}, w_{ac2}) = \\ = \binom{w_{ac1}}{\frac{w_{ac1}+w_{ac2}-w_{ec}}{2}} \binom{Q-w_{ac1}}{\frac{w_{ec}-w_{ac1}+w_{ac2}}{2}} \binom{Q}{w_{ac2}}^{-1} \quad (11)$$

for $w_{ec} = w_{ac1} - w_{ac2} + 2v$ and $v = 0, \ldots, \min(Q - w_{ac1}, w_{ac2})$. The proof can be found in the appendix.

The probability mass function of the extrinsic message is then computed as

$$p(w_{ec}) = \sum_{w_{ac1}=0}^{Q} \sum_{w_{ac2}=0}^{Q} \Big\{ p(w_{ec}|w_{ac1}, w_{ac2}) \\ p(w_{ac1})p(w_{ac2}) \Big\}, \quad (12)$$

where $p(w_{ac1})$ and $p(w_{ac2})$ denote the probability mass functions of the weights of the two inputs.

For check nodes of higher degrees, the probability mass function of the extrinsic message can then be computed recursively by applying (11) and (12) as shown in [3].

V. Decoding Thresholds and Simulations

Using density evolution as developed in Section IV, we are able to determine the decoding threshold of the BVMP decoder as a function of the binary vector message length Q. In the following we will assume that the codewords are transmitted over an AWGN channel. As an example, we computed the threshold for a regular LDPC code of rate $R = 0.5$ with $d_v = 3$ and $d_c = 6$. The results are shown in Fig. 1 and Table I. For $Q = 1$ we obtain the threshold of the binary message passing decoder and for large Q the decoding threshold of the BVMP decoder converges to the decoding threshold of the SPA.

It can be observed that the improvement in decoding performance due to an increased vector length is large for small values of Q. Therefore, the proposed decoding algorithm is attractive for small to moderate Q. For example, increasing Q from one to five leads to a gain of 1.26dB for this specific code.

In order to verify our results, we performed bit error rate simulations of an LDPC code with $d_v = 3$ and $d_c = 6$ for $Q = 1, 2, 3, 5, 10$, and we compared the performance with the SPA. The code was constructed using the PEG algorithm [12] and has a block length $N = 10^3$.

The conversion from weights to L-values as described in Eqn.(1) is pre-computed for every iteration using the weight distributions obtained by density evolution. The thresholds are shown in Table I and the corresponding bit error rate simulations are shown in Fig. 2. The positions of the waterfall regions match up well with our analytical derivation of the decoding threshold, and the slopes of the threshold do not depend on the parameter Q. The small difference between the simulations and the analytical results can be explained by the relatively short block length ($N = 10^3$) whereas the theoretical results hold only in the asymptotic case of infinite block lengths.

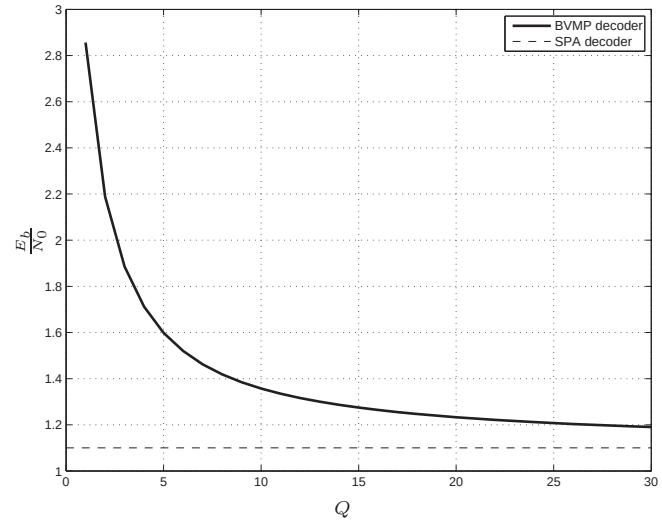

Fig. 1. Decoding threshold in E_b/N_0 as a function of Q for a regular LDPC code with $d_v = 3$ and $d_c = 6$.

TABLE I
Decoding thresholds of a regular LDPC code with $d_v = 3$ and $d_c = 6$.

Q	E_b/N_0 [dB]
1	2.86
2	2.19
3	1.89
5	1.60
10	1.36
SPA	1.10

To assess the decoding complexity of the algorithm, we show the average number of decoding iterations in Fig. 3 where the maximum number of iterations was set to 100. Note that in one iteration the whole binary vector is exchanged between variable and check nodes. Since the complexity of both variable and check node decoder grows linearly with the vector length Q, the overall decoding complexity is proportional to the product of the average number of iterations and the vector length Q. Also the size of the lookup table for the conversion from weights to L-values grows linearly with Q. A future research topic is to adapt the vector length Q during the iterative decoding process in order to minimize the overall decoding complexity.

VI. Summary

In this paper a generalization of a binary message passing algorithm has been introduced, where the variable node decoder is allowed to send binary messages to the check node decoder more than once per iteration. For convenience of analysis, those binary messages have been combined to

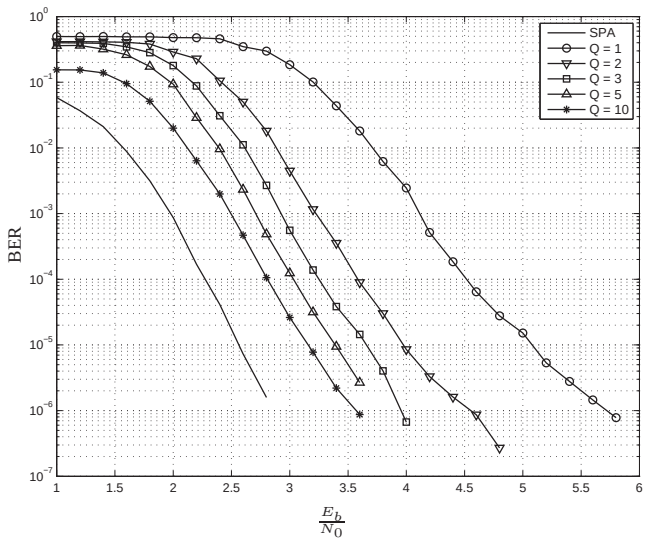

Fig. 2. Bit error rate simulations of a regular LDPC code of length $N = 10^3$ with $d_v = 3$ and $d_c = 6$ for $Q = 1, 2, 3, 5, 10$.

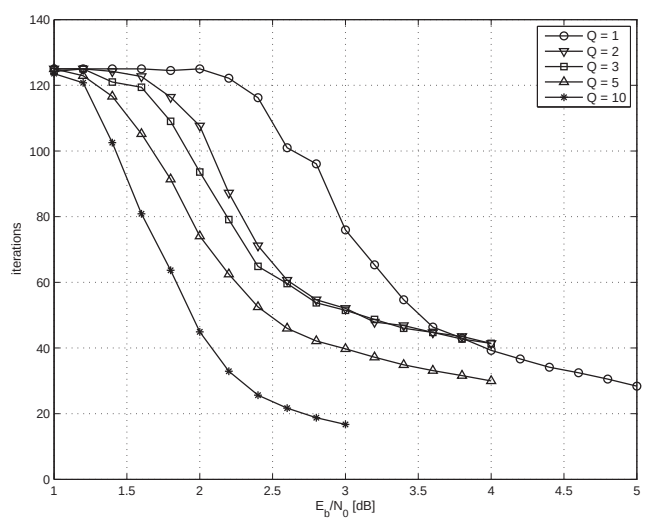

Fig. 3. Average number of decoding iterations of a regular LDPC code of length $N = 10^3$ with $d_v = 3$ and $d_c = 6$ for $Q = 1, 2, 3, 5, 10$. (One iteration refers to the exchange of the whole binary vector message.)

binary vectors, which gives the proposed decoding algorithm its name, *binary vector message passing*. Density evolution for this new decoding algorithm has been developed, and decoding thresholds as a function of the length of the binary vector messages have been determined. Furthermore, a real decoder for finite-length codes has been implemented, and error-rate simulations have been performed. The theoretical analysis as well as the simulation results show a significant gain in performance by increasing the message length, in particular when the initial message length is very small.

In future research we will further analyze the relation between our proposed algorithm and the stochastic decoding approaches from literature. Our approach may provide a new point of view to stochastic decoding, and ideas from stochastic decoding may further improve our method.

Acknowledgment

This work has been supported in parts by the Australian Research Council under ARC Discovery Grant DP0663567 and the ARC Communications Research Network (ACoRN) RN0459498; by the European Commission in the framework of the FP7 Network of Excellence in Wireless COMmunications NEWCOM++ (contract n. 216715); and by the STREP project No. IST-026905 (MASCOT) within the sixth framework programme.

The Telecommunications Research Center Vienna (ftw.) is supported by the Austrian Government and the City of Vienna within the competence center program COMET.

Appendix

Conditional Weight Probability

In this appendix we prove (11) with a combinatorial approach. For convenience, we first formulate the problem again with a simplified notation.

Consider three random binary vectors $\boldsymbol{b}_1, \boldsymbol{b}_2, \boldsymbol{b}_3$ of length Q, and denote w_1, w_2, w_3 their respective Hamming weights. Assume that $\boldsymbol{b}_1$ and $\boldsymbol{b}_2$ are independent and uniformly distributed for any given w_1 and w_2, and that $\boldsymbol{b}_3 = \boldsymbol{b}_1 \oplus \boldsymbol{b}_2$. Assume that $w_1 \geq w_2$. We show now that

$$p(w_3|w_1, w_2) = \binom{w_1}{\frac{w_1+w_2-w_3}{2}} \binom{Q-w_1}{\frac{-w_1+w_2+w_3}{2}} \binom{Q}{w_2}^{-1}$$

for $w_3 \in \mathbb{W}_3$, and that $p(w_3|w_1, w_2) = 0$ otherwise, where

$$\mathbb{W}_3 := \big\{w_1 - w_2 + 2v : v \in \{0, 1, \ldots, \min(Q - w_1, w_2)\}\big\}.$$

Assume that $\boldsymbol{b}_1$ has Hamming weight w_1. Define the index set $\mathcal{A}_0$ with the zero positions in $\boldsymbol{b}_1$, and the index set $\mathcal{A}_1$ with the one positions in $\boldsymbol{b}_1$. Assume that $\boldsymbol{b}_2$ has Hamming weight w_2. For convenience, denote $\mathbb{B}$ the set of all vectors of Hamming weight w_2; thus we have $\boldsymbol{b}_2 \in \mathbb{B}$. We partition $\mathbb{B}$ into subsets $\mathbb{B}_v$, $v = 0, 1, \ldots, \min(Q - w_1, w_2)$; these subsets are chosen such that each vector in $\mathbb{B}_v$ has v ones with indices in $\mathcal{A}_0$ and $(w_2 - v)$ ones with indices in $\mathcal{A}_1$.

Consider now the weight w_3 of $\boldsymbol{b}_3$. For all $\boldsymbol{b}_2 \in \mathbb{B}_v$, the weight of $\boldsymbol{b}_3$ is apparently

$$w_3 = w_1 - w_2 + 2v \tag{13}$$

Therefore, since $\mathbb{B}_v$ partitions $\mathbb{B}$, the possible weights w_3 for given w_1 and w_2 are in the set

$$\mathbb{W}_3 := \big\{w_1 - w_2 + 2v : v \in \{0, 1, \ldots, \min(Q - w_1, w_2)\}\big\}.$$

Correspondingly, all $w_3 \notin \mathbb{W}_3$ have zero probability. On the other hand, if w_3 has a certain value (for given w_1 and w_2), the value of v is fixed by (13), and we know that $\boldsymbol{b}_2 \in \mathbb{B}_v$.

As by assumption, $\boldsymbol{b}_1$ and $\boldsymbol{b}_2$ are independent and uniformly distributed for given w_1 and w_2, we have

$$p(w_3|w_1, w_2) = \frac{|\mathbb{B}_v|}{|\mathbb{B}|} \tag{14}$$

for $w_3 \in \mathbb{W}_3$ and $v = (-w_1 + w_2 + w_3)/2$. The set $\mathbb{B}$ contains all vectors of weight w_2, and thus has size $\binom{Q}{w_2}$. The set $\mathbb{B}_v$ contains all vectors that have v ones with indices in $\mathcal{A}_0$ and $(w_2 - v)$ ones with indices in $\mathcal{A}_1$, and thus $\mathbb{B}_v$ has the size $\binom{Q-w_1}{v}\binom{w_1}{w_2-v}$. The above reasoning is independent of the choice of $\boldsymbol{b_1}$ (provided that its Hamming weight is w_1). This completes the proof.

REFERENCES

[1] R. Gallager, "Low-density parity-check codes," *IEEE Trans. Inform. Theory*, vol. 8, no. 1, pp. 21–28, Jan. 1962.

[2] M. C. Davey and D. MacKay, "Low-density parity-check codes over GF(q)," *IEEE Commun. Lett.*, vol. 2, no. 6, pp. 165–167, Jun. 1998.

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

[4] A. Ashikhmin, G. Kramer, and S. ten Brink, "Extrinsic information transfer functions: model and erasure channel properties," *IEEE Trans. Inform. Theory*, vol. 50, no. 11, pp. 2657–2673, 2004.

[5] T. Richardson, M. Shokrollahi, and R. Urbanke, "Design of capacity-approaching irregular low-density parity-check codes," *IEEE Trans. Inform. Theory*, vol. 47, no. 2, pp. 619–637, Feb. 2001.

[6] T. J. Richardson and R. L. Urbanke, "The capacity of low-density parity-check codes under message-passing decoding," *IEEE Trans. Inform. Theory*, vol. 47, no. 2, pp. 599–618, Feb. 2001.

[7] G. Lechner, T. Pedersen, and G. Kramer, "EXIT chart analysis of binary message-passing decoders," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Nice, France, Jun. 2007.

[8] C. Winstead, V. C. Gaudet, A. Rapley, and C. Schlegel, "Stochastic iterative decoders," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Adelaide, Australia, Sep. 2005.

[9] S. S. Tehrani, W. J. Gross, and S. Mannor, "Stochastic decoding of LDPC codes," *IEEE Commun. Lett.*, vol. 10, no. 10, pp. 716–718, Oct. 2006.

[10] S. S. Tehrani, S. Mannor, and W. J. Gross, "An area-efficient FPGA-based architecture for fully-parallel stochastic LDPC decoding," in *Proc. IEEE Workshop on Signal Processing Systems*, 17-19 Oct. 2007, pp. 255–260.

[11] M. Ardakani and F. Kschischang, "Properties of optimum binary message-passing decoders," *IEEE Trans. Inform. Theory*, vol. 51, no. 10, pp. 3658–3665, Oct. 2005.

[12] X.-Y. Hu, E. Eleftheriou, and D. Arnold, "Regular and irregular progressive edge-growth tanner graphs," *IEEE Trans. Inform. Theory*, vol. 51, no. 1, pp. 386–398, Jan 2005.