

# Exact Split Information Function for SPC

Y. Min\*, F. C. M. Lau\*† and C. K. Tse\*

\*Department of Electronic and Information Engineering, Hong Kong Polytechnic University, Hong Kong

†The Hong Kong Polytechnic University Shenzhen Research Institute

Email: encmlau@polyu.edu.hk, Fax: +852 2362 8439, Tel: +852 2766 6206

**Abstract**—Split information functions are used in deriving closed-form Extrinsic Information Transfer (EXIT) curves of super-variable nodes (SVNs) in doubly-generalized low-density parity-check (DGLDPC) codes. In this letter, we derive an exact split information function for single-parity-check (SPC) codes. The function is very easy to compute and has been verified against the results obtained using the traversal method.

**Index Terms**—Doubly generalized LDPC codes, EXIT curve, split information function.

## I. INTRODUCTION

Low-density parity-check (LDPC) codes make use of repetition codes at the variable nodes and single-parity-check (SPC) codes at the check nodes. Doubly-generalized LDPC (DGLDPC) codes are formed when the repetition codes and the SPC codes are replaced by more complex linear block codes. Subsequently, the nodes are called “super-variable nodes” (SVNs) and “super-check nodes” (SCNs). Researchers have proposed using various types of constituent codes such as Hamming codes and BCH codes in the generalized LDPC and DGLDPC codes [1]–[7].

Similar to the standard LDPC decoders, the iterative decoder of DGLDPC codes can be regarded as two concatenated component codes, including the super-variable-node (SVN) decoder and the super-check-node (SCN) decoder, as shown in Figure 1. For an  $m_a \times n_a$  adjacency matrix, the corresponding DGLDPC code has  $m_a$  SCN decoders and  $n_a$  SVN decoders which are connected by an edge-interleaver.

In each iteration, each SVN decoder takes the *channel information*  $C$  and the *a priori* information  $A_{svn}$  as the input, and then outputs the *extrinsic* information  $E_{svn}$ .  $E_{svn}$ , after passing through the edge-interleaver, becomes the *a priori* information  $A_{scn}$  of the neighboring SCN decoder. Based on  $A_{scn}$ , each SCN decoder generates the *extrinsic* information  $E_{scn}$  and passes it, via the edge-interleaver, back to the SVN decoder as the *a priori* information  $A_{svn}$ . Consequently, two types of channels, namely *communication channel* and *extrinsic channel*, exist in the decoder model.

In [8], ten Brick has proposed using Extrinsic Information Transfer (EXIT) charts to analyze the convergence behavior of turbo codes. Later, the principle of EXIT charts has been successfully applied to study other iteratively-decoded codes such as parallel concatenated codes (PCCs) [9], serially concatenated codes (SCC) [10], convolutional codes [11], LDPC codes [9], [12], repeat-accumulate codes [13], generalized LDPC codes and DGLDPC codes [7], [14], [15].

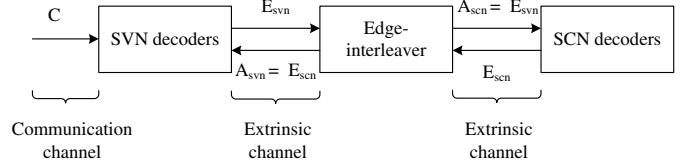


Fig. 1. An iterative decoder of DGLDPC codes.

In [16], it has been shown that codes with capacity-approaching performance over the binary erasure channel (BEC) can be designed by matching exactly the EXIT curves of the component codes. Empirically, this approach can also be applied to the more general cases, i.e., the binary-input additive-white-Gaussian-noise (BI-AWGN) channel. While the closed-form EXIT functions of LDPC codes over both the BEC and the BI-AWGN channel have been derived already, they cannot be applied to the analysis of DGLDPC directly. One common method is to use Monte Carlo simulations to estimate the EXIT charts of DGLDPC codes. However, intensive simulations have to be performed and are very time-consuming.

Another way is to derive the closed-form EXIT functions for all types of SVNs and SCNs in the DGLDPC codes over the BEC. Assume that the *communication channel* and the *extrinsic channel* are BECs and the corresponding erasure probabilities of these two channels are denoted by  $q$  and  $p$ , respectively. For a DGLDPC code, the closed-form EXIT functions of any SVN type and any SCN type over the BEC have been derived in [16]. For a general  $(n_{svn}, k_{svn})$  constituent code with code length  $n_{svn}$  and  $k_{svn}$  information bits used at the SVN, the closed-form EXIT function of this SVN over the BEC is expressed by [16]

$$\begin{aligned}
& I_{e,svn}^{BEC}(p, q) \\
&= 1 - \frac{1}{n_{svn}} \sum_{t=0}^{n_{svn}-1} \sum_{z=0}^{k_{svn}} p^t (1-p)^{n_{svn}-t-1} q^z (1-q)^{k_{svn}-z} \\
&\quad \times [(n_{svn}-t)\tilde{e}_{n_{svn}-t,k_{svn}-z} - (t+1)\tilde{e}_{n_{svn}-t-1,k_{svn}-z}]
\end{aligned} \tag{1}$$

where  $\tilde{e}_{g,h}$  is named as the  $(g, h)$ -th split information function. Moreover,  $\tilde{e}_{g,h}$  is defined as the summation of the ranks of all the possible sub-matrices (denoted by  $S_{g,h}$ ) obtained by choosing  $g$  columns in the corresponding generator matrix with the size of  $k_{svn} \times n_{svn}$  and  $h$  columns in the correspond-

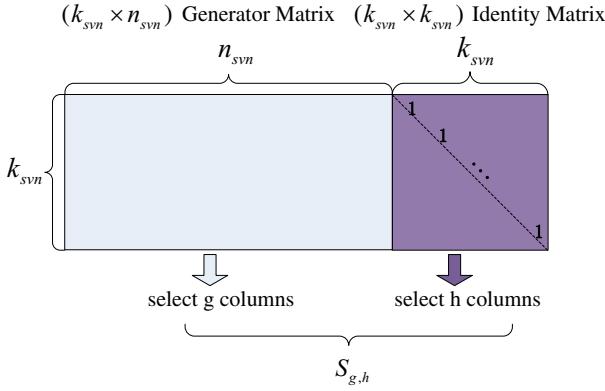


Fig. 2. The definition of the sub-matrix  $S_{g,h}$  for calculating the split information function  $\tilde{e}_{g,h}$  for a general  $(n_{svn}, k_{svn})$  SVN.

ing  $k_{svn} \times k_{svn}$  identity matrix. Figure 2 depicts the definition of the sub-matrix  $S_{g,h}$  for calculating the split information function  $\tilde{e}_{g,h}$ . For the  $(n_{svn}, 1)$  repetition code, (1) leads to

$$I_{e,svn}^{BEC}(p, q) = 1 - qp^{n_{svn}-1} \quad (2)$$

which is the well-known closed-form EXIT function for a variable node with degree- $n_{svn}$  in an LDPC code over the BEC.

Evaluating the  $(g, h)$ -th split information functions  $\tilde{e}_{g,h}$  of all constituent codes at the SVNs can be very time-consuming if closed-form formulas are not available. For a general  $(n, k)$  block code, the number of possible sub-matrices are up to  $2^{n+k}$  for computing  $\tilde{e}_{g,h}$ . Taking the  $(8, 7)$  SPC code used at the SVN as an example, in order to obtain the  $(g, h)$ -th split information function  $\tilde{e}_{g,h}$ , we should calculate the ranks of  $2^{15}$  possible sub-matrices and add them up. In this paper, we derive an exact formula for calculating  $\tilde{e}_{g,h}$  of the SPC codes at SVNs.

## II. SPLIT INFORMATION FUNCTIONS FOR SPC CODES

*Lemma 1:* Assume that

$$\begin{aligned} \mathbf{a} &= (a_1, a_2, \dots, a_i, \dots, a_k) \in \{0, 1\}^k \\ \mathbf{b} &= (b_1, b_2, \dots, b_i, \dots, b_k) \in \{0, 1\}^k \\ \mathbf{c} &= (c_1, c_2, \dots, c_i, \dots, c_k) \in \{0, 1\}^k \end{aligned}$$

and

$$c_i = \text{OR}(a_i, b_i)$$

where the  $\text{OR}(x, y)$  operator is defined as

$$\text{OR}(x, y) = \begin{cases} 0 & \text{if } x = y = 0 \\ 1 & \text{otherwise} \end{cases}.$$

Then the number of possible combinations  $(\mathbf{a}, \mathbf{b})$  satisfying

$$\begin{cases} \sum c_i = t \\ \sum a_i = g \\ \sum b_i = h \end{cases} \quad (3)$$

equals

$${}_k C_t \cdot {}_t C_g \cdot {}_g C_{g+h-t}, \quad (4)$$

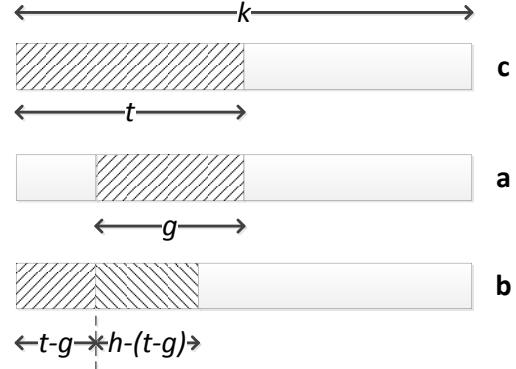


Fig. 3. The vectors  $\mathbf{c}$ ,  $\mathbf{a}$  and  $\mathbf{b}$  shown from top to bottom. The shadowed parts in the vectors indicate the positions of ‘1’s. There are  $t$ ,  $g$  and  $h$  ‘1’s in the vectors  $\mathbf{c}$ ,  $\mathbf{a}$  and  $\mathbf{b}$ , respectively.

where  $0 \leq g, h \leq k$ ;  $\max(g, h) \leq t \leq \min(g + h, k)$ ; and  ${}_k C_t = \frac{k!}{(k-t)!t!}$ .

*Proof:* Referring to Fig. 3, each of the vectors  $\mathbf{a}$ ,  $\mathbf{b}$  and  $\mathbf{c}$  contains  $k$  elements. Moreover,  $t$  of the elements in  $\mathbf{c}$  are ‘1’s. First we select  $t$  out of the  $k$  locations to put the ‘1’s, and there are  ${}_k C_t$  combinations. Among these  $t$  locations,  $g$  of them coincide with the locations where the elements of  $\mathbf{a}$  also equal ‘1’. There are  ${}_t C_g$  such choices. Among the  $h$  ‘1’s in  $\mathbf{b}$ ,  $(t-g)$  of them do not overlap (in terms of location) with those in  $\mathbf{a}$ . The remaining  $(h-(t-g)) = (g+h-t)$  ‘1’s in  $\mathbf{b}$  overlap with the ‘1’s in  $\mathbf{a}$ . The number of combinations is  ${}_g C_{g+h-t}$ . Therefore, the overall number of combinations satisfying (3) equals  ${}_k C_t \cdot {}_t C_g \cdot {}_g C_{g+h-t}$ .  $\square$

Denote  $\mathbf{I}_k$  as the  $k \times k$  identity matrix, and  $\mathbf{e}_k$  as the all-one vector of size  $k \times 1$ . The generator matrix  $\mathbf{G}$  of the  $(n, n-1)$  SPC code is thus written as

$$\mathbf{G} = [\mathbf{I}_k, \mathbf{e}_k]$$

where  $k = n - 1$ . The definition of the sub-matrix  $S_{g,h}$  for calculating the split information function  $\tilde{e}_{g,h}$  for the  $(n, n-1)$  SPC code is shown in Fig. 4.

In order to calculate the split information function of the  $(n, n-1)$  SPC code, we need to select  $g$  columns from  $\mathbf{G}$  (the LHS matrix in Fig. 4) and  $h$  columns from  $\mathbf{I}_k$  (the RHS matrix in Fig. 4) to construct a sub-matrix  $S_{g,h}$ . Let  $\text{Rank}(\mathbf{W})$  denote the rank of the matrix  $\mathbf{W}$ . Then the question is how to get the sum of  $\text{Rank}(S_{g,h})$  for all cases to obtain  $\tilde{e}_{g,h}$ .

*Theorem 1:* The  $\tilde{e}_{g,h}$  defined in Problem 1 is given by

$$\begin{aligned} \tilde{e}_{g,h} &= \sum_{t=\max(g,h)}^{\min(g+h,n-1)} t \cdot {}_{n-1} C_t \cdot {}_t C_g \cdot {}_g C_{g+h-t} \\ &+ \sum_{t=\max(g-1,h)}^{\min(g+h-1,n-1)} [\min(t+1, n-1) \\ &\quad \cdot {}_{n-1} C_t \cdot {}_t C_{g-1} \cdot {}_{g-1} C_{g+h-1-t}]. \end{aligned} \quad (5)$$

*Proof:* We consider two cases: (a)  $\mathbf{e}_k$  in  $\mathbf{G}$  is not selected

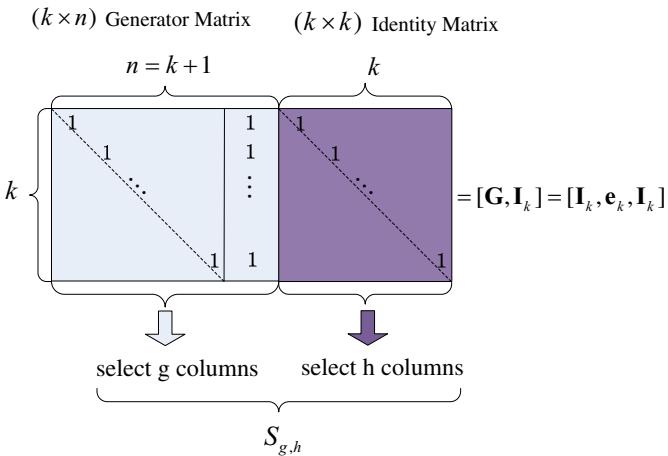


Fig. 4. The definition of the sub-matrix  $S_{g,h}$  for calculating the split information function  $\tilde{e}_{g,h}$  for the  $(n, n-1)$  SPC code.

to form  $S_{g,h}$ ; (b)  $e_k$  in  $G$  is selected to form  $S_{g,h}$ .

Case-(a):  $e_k$  in  $G$  is not selected. In this case, in order to construct the sub-matrix  $S_{g,h}$  of size  $k \times (g+h)$ , we only need to select  $g$  and  $h$  columns from the two identity matrices  $I_k$ , respectively. We denote

$$a_i = \begin{cases} 1 & \text{if the } i\text{-th column of the first } I_k \text{ is selected} \\ 0 & \text{otherwise,} \end{cases}$$

$$b_i = \begin{cases} 1 & \text{if the } i\text{-th column of the second } I_k \text{ is selected} \\ 0 & \text{otherwise.} \end{cases}$$

Then

$$\text{Rank}(S_{g,h}) = \sum c_i = \sum \text{OR}(a_i, b_i), \quad (6)$$

where  $\sum a_i = g$  and  $\sum b_i = h$ . It is easy to see that

$$\max(g, h) \leq \text{Rank}(S_{g,h}) \leq \min(g+h, k) = \min(g+h, n-1). \quad (7)$$

Based on Lemma 1, the number of combinations for selecting these columns such that  $\text{Rank}(S_{g,h}) = t$  is

$${}_{n-1}C_t \cdot {}_tC_g \cdot {}_gC_{g+h-t}. \quad (8)$$

Hence the sum of the ranks for all the possible cases given in (7) is given by

$$\sum_{t=\max(g,h)}^{\min(g+h,n-1)} t \cdot {}_{n-1}C_t \cdot {}_tC_g \cdot {}_gC_{g+h-t}. \quad (9)$$

Case-(b):  $e_k$  in  $G$  is selected. First we select  $g-1$  columns and  $h$  columns from the two identity matrices  $I_k$ , respectively, to construct  $S'_{g,h}$  of size  $k \times (g-1+h)$ . From Case-(a), we know that

$$\begin{aligned} \max(g-1, h) &\leq \text{Rank}(S'_{g,h}) \leq \min(g-1+h, k) \\ &= \min(g-1+h, n-1) \end{aligned} \quad (10)$$

TABLE I  
THE SPLIT INFORMATION FUNCTION OF  $(5, 4)$  SPC CODE.

$g$	$h$	$\tilde{e}_{g,h}$	$g$	$h$	$\tilde{e}_{g,h}$
0	0	0	3	0	30
0	1	4	3	1	136
0	2	12	3	2	222
0	3	12	3	3	156
0	4	4	3	4	40
1	0	5	4	0	20
1	1	36	4	1	80
1	2	78	4	2	120
1	3	68	4	3	80
1	4	20	4	4	20
2	0	20	5	0	4
2	1	104	5	1	16
2	2	192	5	2	24
2	3	148	5	3	16
2	4	40	5	4	4

and the number of combinations equals

$${}_{n-1}C_{\text{Rank}(S'_{g,h})} \cdot {}_{\text{Rank}(S'_{g,h})}C_{g-1} \cdot {}_{g-1}C_{g-1+h-\text{Rank}(S'_{g,h})}.$$

Noting that when we combine  $S'_{g,h}$  and  $e_k$  to form  $S_{g,h}$ , the rank of  $S_{g,h}$  is given by

$$\text{Rank}(S_{g,h}) = \min(\text{Rank}(S'_{g,h}) + 1, n-1). \quad (11)$$

The summation of all  $\text{Rank}(S_{g,h})$  for Case-(b) is therefore

$$\sum_{t=\max(g-1,h)}^{\min(g+h-1,n-1)} \min(t+1, n-1) \cdot {}_{n-1}C_t \cdot {}_tC_g \cdot {}_gC_{g+h-1-t}. \quad (12)$$

Combining the two cases together, we obtain the result in (5).  $\square$

In Table I and Table II, we show our numerical results  $\tilde{e}_{g,h}$  calculated by Theorem 1 for  $(5, 4)$  SPC code and  $(6, 5)$  SPC code, respectively. Our numerical results  $\tilde{e}_{g,h}$  calculated by Theorem 1 are the same as those evaluated by considering all possible combinations of the sub-matrices and their ranks.

### III. CONCLUSION

We have derived an exact formula for calculating the split information function for SPC codes used at the SVNs of DGLDPC codes. The results have further been verified against those found by the traversal method (i.e., forming all possible sub-matrices, evaluating their ranks and summing them up). The exact formula can facilitate DGLDPC designers evaluating the closed-form EXIT function of SVN over the BEC and subsequently over the BI-AWGN channel.

### ACKNOWLEDGEMENTS

The work described in this paper was partially supported by a grant from the RGC of the Hong Kong SAR, China (Project No. PolyU 519011) and by the National Natural Science Foundation of China (Grant No. 61372095).

TABLE II  
THE SPLIT INFORMATION FUNCTION OF (6, 5) SPC CODE.

$g$	$h$	$\tilde{e}_{g,h}$	$g$	$h$	$\tilde{e}_{g,h}$	$g$	$h$	$\tilde{e}_{g,h}$
0	0	0	3	0	60	6	0	5
0	1	5	3	1	350	6	1	25
0	2	20	3	2	800	6	2	50
0	3	30	3	3	890	6	3	50
0	4	20	3	4	480	6	4	25
0	5	5	3	5	100	6	5	5
1	0	6	4	0	60			
1	1	55	4	1	325			
1	2	160	4	2	690			
1	3	210	4	3	720			
1	4	130	4	4	370			
1	5	30	4	5	75			
2	0	30	5	0	30			
2	1	200	5	1	150			
2	2	500	5	2	300			
2	3	600	5	3	300			
2	4	345	5	4	150			
2	5	75	5	5	30			

## REFERENCES

- [1] M. Lentmaier and K. Zigangirov, "On generalized low-density parity-check codes based on Hamming component codes," *IEEE Commun. Letters*, vol. 3, no. 8, pp. 248–250, Aug. 1999.
- [2] G. Liva and W. E. Ryan, "Short low-error-floor Tanner codes with Hamming nodes," in *Proc. IEEE Milcom*, pp. 208–213, Oct. 2005.
- [3] N. Miladinovic and M. Fossorier, "Generalized LDPC codes with Reed-Solomon and BCH codes as component codes for binary channels," in *Proc. IEEE GLOBECOM*, vol. 3, Nov. 2005, p. 6.
- [4] I. Djordjevic, L. Xu, T. Wang, and M. Cvijetic, "GLDPC codes with Reed-Muller component codes suitable for optical communications," *IEEE Commun. Letters*, vol. 12, no. 9, pp. 684–686, Sep. 2008.
- [5] A. S. Shadi, G. Liva, and W. E. Ryan, "Low-floor Tanner codes via Hamming-node or RSCLC-Node doping," in *Proc. the 16th international conference on Applied Algebra, Algebraic Algorithms and Error-Correcting Codes*, 2006, pp. 245–254.
- [6] J. Chen and R. M. Tanner, "A hybrid coding scheme for the Gilbert-Elliott channel," *IEEE Trans. Commun.*, vol. 54, pp. 1787–1796, 2006.
- [7] Y. Wang and M. Fossorier, "Doubly generalized LDPC codes over the AWGN channel," *IEEE Trans. Commun.*, vol. 57, no. 5, pp. 1312–1319, May 2009.
- [8] S. ten Brink, "Convergence of iterative decoding," *Electronics Letters*, vol. 35, pp. 806–808, 1999.
- [9] S. ten Brink, G. Kramer, and A. Ashikhmin, "Convergence behavior of iteratively decoded parallel concatenated codes," *IEEE Trans. Commun.*, vol. 49, pp. 1727–1737, Oct. 2001.
- [10] S. ten Brink, "Design of serially concatenated codes based on iterative decoding convergence," in *Proc. the 2nd International Symposium on Turbo Codes and Related Topics*, Sep. 2000, pp. 319–322.
- [11] M. Tuchler, "Design of serially concatenated systems depending on the blocklength," *IEEE Trans. Commun.*, vol. 52, no. 2, pp. 209–218, Feb. 2004.
- [12] S. ten Brink, G. Kramer, and A. Ashikhmin, "Design of Low-Density Parity-Check codes for modulation and detection," *IEEE Trans. Commun.*, vol. 52, no. 4, pp. 670–1737, Apr. 2004.
- [13] S. ten Brink and G. Kramer, "Design of repeat-accumulate codes for iterative detection and decoding," *IEEE Trans. on Signal Processing*, vol. 51, no. 11, pp. 2764–2772, Nov. 2003.
- [14] G. Yue, L. Ping, and X. Wang, "Generalized low-density parity-check codes based on Hadamard constraints," *IEEE Trans. Inform. Theory*, vol. 53, no. 3, pp. 1058–1079, Mar. 2007.
- [15] E. Paolini, M. Fossorier, and M. Chiani, "Generalized stability condition for generalized and doubly-generalized LDPC codes," in *Proc. IEEE ISIT*, Jun. 2007, pp. 1536–1540.
- [16] A. Ashikhmin, G. Kramer, and S. ten Brink, "Extrinsic information transfer functions: model and erasure channel properties," *IEEE Trans. Inform. Theory*, vol. 50, pp. 2657–2673, Nov. 2004.

TABLE III  
THE SPLIT INFORMATION FUNCTION OF (7, 6) SPC CODE.

$g$	$h$	$\tilde{e}_{g,h}$	$g$	$h$	$\tilde{e}_{g,h}$	$g$	$h$	$\tilde{e}_{g,h}$
0	0	0	3	0	105	6	0	42
0	1	6	3	1	750	6	1	252
0	2	30	3	2	2175	6	2	630
0	3	60	3	3	3300	6	3	840
0	4	60	3	4	2760	6	4	630
0	5	30	3	5	1200	6	5	252
0	6	6	3	6	210	6	6	42
1	0	7	4	0	140	7	0	6
1	1	78	4	1	930	7	1	36
1	2	285	4	2	2550	7	2	90
1	3	500	4	3	3680	7	3	120
1	4	465	4	4	2940	7	4	90
1	5	222	4	5	1230	7	5	36
1	6	42	4	6	210	7	6	6
2	0	42	5	0	105			
2	1	342	5	1	666			
2	2	1080	5	2	1740			
2	3	1740	5	3	2400			
2	4	1530	5	4	1845			
2	5	696	5	5	750			
2	6	126	5	6	126			