

# On the Construction of Regular QC-LDPC Codes with Low Error Floor

Abdullah Sarıduman and Ali Emre Pusane and Z. Caner Taşkın

**Abstract**—Quasi-cyclic low-density (QC-LDPC) codes benefit from efficient encoding hardware and display excellent error correcting performance. Therefore, they have been accepted into the 5G standards in addition to being potential candidates for next generation mobile systems. However, dominant trapping sets in the Tanner graphs cause some failures for the iterative decoding algorithm. In this paper, we propose a simulated annealing based method to find regular QC-LDPC codes without dominant trapping sets and have good error floor performance. Simulation results show that our proposed method generates QC-LDPC codes that have the best trapping sets distribution among recent works and better frame rate performance than others.

**Index Terms**—Low Density Parity Check (LDPC) codes, trapping sets, quasi-cyclic (QC), elementary trapping sets, error floor.

## I. INTRODUCTION

Protograph-based low density parity-check (LDPC) codes are popular nowadays due to their high error correcting performance [1]. In particular, regular quasi-cyclic LDPC (QC-LDPC) codes are preferred because of their easy implementation and they have already been adopted into many standards [2], [3]. However, especially at high signal-to-noise ratio (SNR) values, QC-LDPC codes have an error floor problem, i.e., their error performance ceases to improve much despite using higher SNR channels. The main cause of this problem are some topological structures in the Tanner graph of the QC-LDPC code. These structures, which Richardson named as trapping sets, are known to cause iterative decoding algorithms to fail [4].

Cycles are the main reason of the trapping sets. Therefore, the absence of small cycles in the Tanner graph reduces the number of dominant trapping sets that causes the error floor. While designing a QC-LDPC code, many studies have focused on the girth value of the Tanner graph, the size of a smallest cycle in the Tanner graph, and the number of the cycles with girth size [5], [6], [7]. Besides them, some studies have tried to directly reduce the number of dominant trapping sets in the Tanner graph, which gives more efficient results [8], [9], [10]. [8] proposed RandPEG-noTS(5,3) algorithm that eliminates (5,3) trapping sets and decreases the number of (6,4) trapping sets. [9] constructed QC-LDPC codes that are free of some small trapping sets by a cycle control algorithm. [10] improved the minimum lifting degree of some base graphs that are free of small trapping sets. In both [11] and [12], algorithms that are capable of exhaustive enumeration of the elementary trapping sets (ETS) for regular LDPC codes were

proposed, and it has paved the way for designing QC-LDPC codes without dominant trapping sets. Moreover, advanced algorithms to exhaustively enumerate ETSs are also appearing in very recent works, such as the one proposed in [13].

In this letter, we aim to construct QC-LDPC codes with much better trapping sets distributions using a simulated annealing algorithm that improves an initially supplied QC-LDPC code. To achieve this, the dominant elementary trapping sets of a QC-LDPC code are roughly enumerated with Algorithm 1 given in [11] and the cost value of this QC-LDPC code is calculated according to its ETS distribution. With the help of the simulated annealing (SA) algorithm, QC-LDPC codes without small ETSs are found.

The remainder of the paper is structured as follows: Section II introduces the basic concepts and notations of QC-LDPC codes. In Section III, the proposed simulated annealing algorithm is explained. Then, Section IV presents construction examples and compares their performance with QC-LDPC codes constructed by other works. Finally, Section V concludes the paper.

## II. PRELIMINARIES

An  $(n, k)$  LDPC code can be represented in terms of a bipartite graph,  $G$  (also called the Tanner graph), with two sets of nodes:  $n$  variable nodes,  $V = \{v_1, \dots, v_n\}$ , and  $m = n - k$  check nodes,  $C = \{c_1, \dots, c_m\}$ . The length of a shortest cycle in  $G$  is called girth,  $g$ . For a variable node subset  $S \subset G$ ,  $G(S)$  denotes the induced subgraph of  $S$ .  $O(S)$  represents the set of check nodes with odd degree in  $G(S)$ . An LDPC code is said to be regular if all check nodes and variable nodes in  $G$  have the same degree.  $d_c$  and  $d_s$  denote the degree of check nodes and variable nodes, respectively.

The parity check matrix of QC-LDPC codes can be presented by a permutation shift matrix  $P$  and a lifting degree  $L$ ,

$$P = \begin{bmatrix} p_{0,0} & p_{0,1} & \dots & p_{0,d_c-1} \\ p_{1,0} & p_{0,1} & \dots & p_{1,d_c-1} \\ \vdots & \vdots & \ddots & \vdots \\ p_{d_s-1,0} & p_{d_s-1,1} & \dots & p_{d_s-1,d_c-1} \end{bmatrix}, \quad (1)$$

The parity check matrix,  $H$ , is constructed from  $I(p_{i,j})$  matrices, that are the  $L \times L$  identity matrices whose elements in each row are cyclically shifted to the left by  $p_{i,j}$ .

For QC-LDPC codes, a cycle of length  $k$  exists in the Tanner graph if and only if

$$\sum_{l=0}^{k-1} (p_{i_l, j_l} - p_{i_{l+1}, j_l}) = 0 \pmod L, \quad (2)$$

where  $i_k = i_0$ ,  $i_l \neq i_{l+1}$ ,  $j_l \neq j_{l+1}$  is satisfied.

**Definition 1.** A variable node set  $T \subset G$  is a  $(a, b)$  trapping set such that  $|T| = a$  and  $|O(T)| = b$ .

A  $(a, b)$  trapping set (TS) is a dominant TS if both  $a$  and  $b$  are small numbers [4]. Also, a trapping set  $T$  is elementary TS if all check nodes have degree one or two in  $G(T)$ . Elementary trapping sets with small number of  $a$  and  $b$  are among the most dominant TS types [4].

### III. SIMULATED ANNEALING ALGORITHM

**Definition 2.** A permutation shift matrix  $P^S$  is a neighbor of the permutation shift matrix  $P^T$  if and only if the girth of the QC-LDPC code generated by  $P^S$  is greater than or equal to the girth of the QC-LDPC code generated by  $P^T$ , and there is exactly one  $p_{i,j}^S \in P^S$  such that

$$p_{i,j}^S \neq p_{i,j}^T \pmod L, \quad (3)$$

where  $i=1, 2, \dots, d_s$ ,  $j=1, 2, \dots, d_c$  and  $p_{i,j}^T \in P^T$ .

**Definition 3.** A set that consists of permutation shift matrices is the neighbor set of the permutation shift matrix  $P^T$  if and only if every permutation shift matrix in the set is a neighbor of the permutation shift matrix  $P^T$ . This set will be represented as  $N(P^T)$  in the rest of the paper.

It is trivial to find the neighbor set of a permutation shift matrix by using (2) and Definition 2. There are exactly  $L \times d_c \times d_s$  candidate permutation shift matrices that could be neighbors of a permutation shift matrix, and they can be checked whether they provide the equality in the (2) or not in polynomial time.

For short-length QC-LDPC codes, [11] and [12] show that all  $(a, b)$  ETS can be found with a simple search algorithm. In this work, Algorithm 1 given in [11] is adopted to find the ETSs. Instead of counting all ETSs, only one variable node is selected from each block, and the cycles and ETSs to which it is connected to are counted. Then, the results are multiplied with  $L$  to take into account the multiplicity that is caused by the quasi-cyclic structure. Therefore, it is sufficient to examine exactly  $d_c$  variable nodes. The disadvantage of this approach is that it causes the ETS distribution to be found roughly instead of exactly, because some of ETSs with the same isomorphic structures are counted multiple times. The advantage is that iteration time is considerably shortened. In the last iteration, however, the ETS distribution of the resulting matrix is exhaustively determined by the Algorithm 1 given in [11].

Using a hill-climbing algorithm, it is possible to improve the ETS distribution of the initially given permutation shift matrix. However, such an algorithm is likely to get stuck in a local optimal point regardless of the selection policy. On the other hand, a simulated annealing algorithm can help converge to a

global optimal solution. According to the gain in the objective value, new solution is accepted or not in the simulation annealing algorithm. If there is positive contribution, the new solution is accepted. Otherwise, Metropolis' criterion based on Boltzman's probability is applied. The acceptance probability is determined with

$$P(A) = e^{-\Delta E/(kT)}, \quad (4)$$

where  $\Delta E$  is the difference between objective function value of the current solution and candidate solution,  $k$  is Boltzmann's constant, and  $T$  is the temperature, respectively.

In the literature, there are various cooling schedules. Geometric cooling rule is one of the simple cooling schedules, and is employed in this work. The update rule is defined as

$$T_{i+1} = \lambda T_i, \quad (5)$$

where  $\lambda$  is the temperature constant that is smaller than 1 but close to 1. The initial temperature is chosen as a big number to accept every solution in the early of iterations. The proposed simulated annealing algorithm is given in Algorithm 2.

---

#### Algorithm 1 Simulated Annealing Algorithm

---

**Input:** Permutation shift matrix  $P^{initial}$ , objective function,  $g(x)$ , max. iteration number  $M$ , initial temperature  $T$ , cooling factor  $\lambda$

---

1.  $P^{initial} \rightarrow P^{cand.}, P^{best.}$
  2.  $g(P^{initial}) \rightarrow min.cost$
  3. **For**  $i = 0 : M$
  4.     Generate  $N(P^{cand.})$
  5.     Randomly select a permutation shift matrix  $P^j$ , from  $N(P^{cand.})$
  6.     Compute the cost  $g(P^j)$
  7.     **if**  $g(P^j) \leq g(P^{cand.})$
  8.          $P^j \rightarrow P^{cand.}$
  9.     **else**
  10.         Compute  $P(A)$  using Equation (4).
  11.         Generate an output  $r$  from a uniform R.V. over  $[0,1]$ .
  12.         **if**  $r \leq P(A)$
  13.              $P^j \rightarrow P^{cand.}$
  14.         **else** Go to STEP 5.
  15.      $T_{i+1} = \lambda T_i$
  16.     **if**  $g(P^{cand.}) < min.cost$
  17.          $P^{cand.} \rightarrow P^{best}$
  18.      $g(P^{cand.}) \rightarrow min.cost$
  19.     **End if**
  19. **End For**
- 

**Output:**  $P^{best.}$

---

Decision variables and decision weights of the objective function are critical to the frame error rate (FER) performance of the QC-LDPC code generated by the simulated annealing algorithm. The number of the dominant ETS in Tanner graph is used as main decision variables in this work. For an  $(a, b)$  ETS class, the number of ETS that are present in the Tanner graph is represented as  $n_{a,b}$ . Objective weights are chosen in proportion to the contribution of ETS class to the error floor. The weight vector with three different decision weights are determined,  $\mathbf{w}=[w_1 \ w_2 \ w_3]$ , according to following rules:

TABLE I  
ETS OF LDPC CODES WITHIN THE RANGE OF  $A \leq 10$  AND  $B \leq 3$

ETS	Tanner	RandPEG-noTS(5,3)	$P_1$
(5,3)	155	-	-
(7,3)	930	62	-
(8,2)	465	-	-
(9,3)	1860	496	155
(10,2)	1395	31	-

- The coefficients of the decision variables are determined as  $w_1$  for the ETS classes ( $S_1$ ) that do not exist in the Tanner graph of the permutation shift matrix ( $P^{initial}$ ) taken as input for the simulated annealing algorithm. By selecting  $w_1$  value too large, the creation of non-existent ETS in the Tanner graph of  $P^{initial}$  is prevented.
- The coefficient of  $w_2$  is used for the ETS classes ( $S_2$ ) which are required to be completely removed in Tanner graph of  $P^{initial}$ . The number of  $w_2$  is chosen to be much smaller than  $w_1$  and greater than  $w_3$ . During the iterations of simulated annealing algorithm, the ETS in this class are allowed to occur sometimes, but has been targeted to be destroyed at the end of algorithm.
- For the ETSs classes ( $S_3$ ), whose numbers in the Tanner graph are desired to be reduced,  $w_3$  coefficients are used.

In this manner, the objective function of the simulated annealing algorithm is given as

$$g(x) = \sum_{(a,b) \in S_1} w_1 n_{a,b} + \sum_{(a,b) \in S_2} w_2 n_{a,b} + \sum_{(a,b) \in S_3} w_3 n_{a,b}. \quad (6)$$

#### IV. SIMULATION RESULTS

In this section, four examples will be presented to analyze the performance of proposed method. All simulation results are based on the sum-product message passing decoding algorithm with maximum number of iterations equal to 100. For each SNR point, fifty frame errors are counted.

**Example 1.** A regular QC-LDPC code with  $d_s=3$ ,  $d_c=5$  and  $L=31$  is constructed. Parameters of simulated annealing algorithm are taken as  $T=1000$ ,  $\lambda=0.99$ ,  $S_1=\{(6,2)\}$ ,  $S_2=\{(5,3), (8,2), (10,2)\}$ ,  $S_3=\{(7,3), (9,3)\}$ ,  $\mathbf{w}=[10^6 \ 10^3 \ 10^0]$ . To generate the input permutation shift matrix, progressive edge algorithm (PEG) given in [5] is used. According to these settings, the permutation shift matrix,  $P_1$ , is generated as

$$P_1 = \begin{bmatrix} 15 & 29 & 17 & 18 & 12 \\ 6 & 21 & 14 & 20 & 2 \\ 23 & 23 & 4 & 13 & 22 \end{bmatrix}. \quad (7)$$

The ETS distributions of QC-LDPC codes constructed by Tanner Codes in [14], RandPEG no(5,3) Algorithm in [8], and  $P_1$  are given in Table I. The results in this table show that the QC-LDPC code generated by the permutation shift matrix  $P_1$

TABLE II  
ETS OF LDPC CODES WITHIN THE RANGE OF  $A \leq 12$  AND  $B \leq 3$

ETS	$C_1$	$P_2$	$C_2$	$P_3$
(9,3)	246	-	252	126
(11,3)	1230	656	2142	1197
(12,2)	123	-	63	-

avoids almost every  $(a, b)$  elementary trapping set that satisfies  $a \leq 10$  and  $b \leq 3$ , except for a few (9,3) ETS.

**Example 2.** A regular QC-LDPC code with  $d_s=3$ ,  $d_c=5$  and  $L=41$  is constructed. Parameters of simulated annealing algorithm are taken as  $T=10000$ ,  $\lambda=0.8$ ,  $S_1=\{(5,3), (6,2), (7,3), (8,2), (10,2)\}$ ,  $S_2=\{(12,2)\}$ ,  $S_3=\{(9,3), (11,3)\}$ ,  $\mathbf{w}=[10^6 \ 10^4 \ 10^0]$ . For the input permutation shift matrix, the matrix  $C_1$  given in [9] is used. According to these settings, the permutation shift matrix,  $P_2$ , is constructed as

$$P_2 = \begin{bmatrix} 0 & 27 & 0 & 16 & 6 \\ 28 & 26 & 5 & 22 & 20 \\ 11 & 11 & 4 & 9 & 16 \end{bmatrix}. \quad (8)$$

The ETS distributions of QC-LDPC codes constructed by the permutation shift matrix  $C_1$ , given in [9], and permutation shift matrix  $P_2$  are presented in Table II. The statistics in this table show that the QC-LDPC code generated by  $P_2$  avoids every  $(a, b)$  trapping set that satisfies  $a \leq 12$  and  $b \leq 2$ .

**Example 3.** A regular QC-LDPC code with  $d_s=3$ ,  $d_c=6$  and  $L=63$  is constructed. Parameters of simulated annealing algorithm are taken as  $T=10000$ ,  $\lambda=0.95$ ,  $S_1=\{(5,3), (6,2), (7,3), (8,2), (10,2)\}$ ,  $S_2=\{(12,2)\}$ ,  $S_3=\{(9,3), (11,3)\}$ ,  $\mathbf{w}=[10^6 \ 10^4 \ 10^0]$ . For the input permutation shift matrix, the matrix  $C_2$  given in [9] is used. According to these settings, the permutation shift matrix,  $P_3$ , is constructed as

$$P_3 = \begin{bmatrix} 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 20 & 8 & 33 & 29 & 59 \\ 0 & 9 & 26 & 24 & 32 & 58 \end{bmatrix}. \quad (9)$$

As in Example 2, ETSs that satisfy  $a \leq 12$  and  $b \leq 2$  are completely removed from the Tanner graph of the proposed permutation shift matrix  $P_3$ . The ETS list of the suggested QC-LDPC codes is given in the Table II.

Figure 1 shows the FER performance of several regular LDPC codes listed in Table I and Table II for binary phase shift keying (BPSK) modulation over the additive white Gaussian noise (AWGN) channel. It can be seen clearly that the suggested QC-LDPC codes have better FER performance compared to the existing QC-LDPC codes in the high SNR region.

**Example 4.** In this example, we improve the lifting degree of the codes with the fully-connected  $3 \times 5$ ,  $3 \times 6$ , and  $3 \times 10$  base graphs given in [9] and [10]. In these studies, the goal is to construct QC-LDPC codes that do not contain leafless ETSs (LETSS) satisfying the conditions  $a \leq 8$ ,  $b \leq 3$  with the smallest possible lifting degree. [9] found the codes with lifting

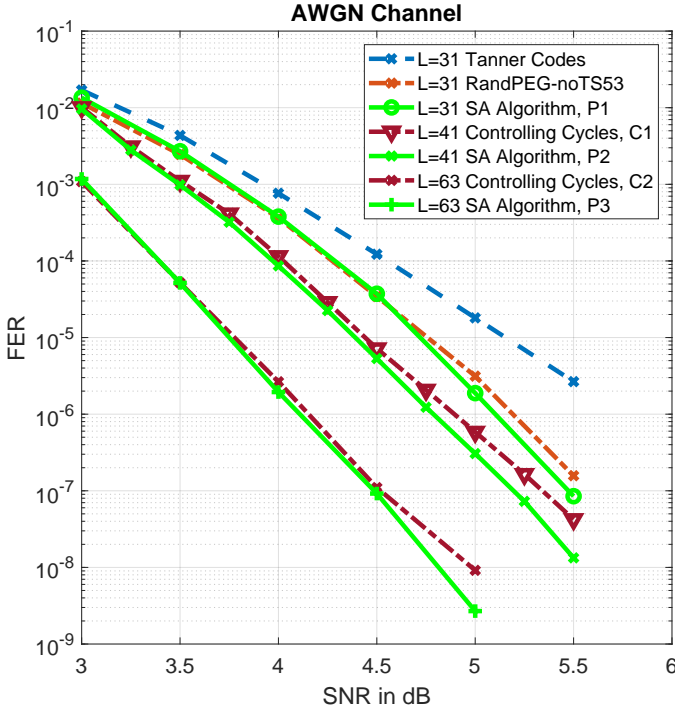


Fig. 1. FER performance of the constructed QC-LDPC codes.

TABLE III  
TIME PERFORMANCE OF THE ALGORITHM 1

	P1	P2	P3	P4	P5	P6
Max. Size of Counted Cycles	14	16	16	10	12	12
Iteration Number	1000	200	50	147	262	47
Time	3h	23h	75h	0.1h	9h	25h

degrees 41, 61, and 181, for the three base graphs, respectively. [10] improved these values as 27, 41, and 165, respectively. In this work, QC-LDPC codes with minimum lifting values 24, 39, and 152 are found by Algorithm 1. Permutation matrices with the minimum lifting degrees of these codes are obtained as

$$P_4 = \begin{bmatrix} 11 & 16 & 3 & 13 & 20 \\ 11 & 2 & 19 & 12 & 17 \\ 3 & 16 & 0 & 16 & 13 \end{bmatrix}, \quad (10)$$

$$P_5 = \begin{bmatrix} 33 & 5 & 32 & 30 & 29 & 22 \\ 15 & 14 & 1 & 33 & 10 & 34 \\ 23 & 4 & 38 & 0 & 15 & 6 \end{bmatrix}, \quad (11)$$

$$P_6 = \begin{bmatrix} 131 & 2 & 70 & 65 & 80 & 101 & 116 & 14 & 116 & 10 \\ 41 & 87 & 119 & 119 & 68 & 73 & 54 & 48 & 4 & 133 \\ 3 & 12 & 150 & 147 & 11 & 5 & 33 & 114 & 27 & 133 \end{bmatrix}. \quad (12)$$

Except for the cooling factor  $\lambda$ , the parameters of the simulated annealing algorithm are taken for these three base graphs as  $T = 100$ ,  $S_1 = \{(6, 2)\}$ ,  $S_2 = \{(5, 3), (8, 2)\}$ ,  $S_3 = \{(7, 3)\}$ ,  $\mathbf{w} = [10^3 \ 10^1 \ 10^0]$ .  $\lambda$  values are taken as 0.9, 0.9, and 0.8, respectively. To generate the input permutation shift matrices, PEG algorithm is used for each base graph.

Although it is an offline task to produce a permutation matrix, the construction time of the permutation matrices

given in the examples is presented in Table III to demonstrate the complexity of the proposed algorithm. The algorithm ran on a desktop computer with 2GHz CPU and 46 GB RAM. The selected parameters of the SA algorithm determine the construction times of the  $P_1$ ,  $P_2$ , and  $P_3$  matrices. The algorithm continues even if it achieves the best result in a certain iteration. On the other hand, for the construction of the  $P_4$ ,  $P_5$ , and  $P_6$  matrices, the algorithm is terminated when the objective function value is 0, i.e., when the desired target is reached.

Cycle counting is the most time and memory consuming part of the algorithm. However, in practice it is quite manageable for small-size cycles. Therefore, QC-LDPC codes without small-size TSs can be generated with Algorithm 1 even for large  $d_c$  and  $L$  values, whereas generation of large QC-LDPC codes with relatively large-sized TSs optimized is not possible. However, since small TSs are more dominant, the proposed algorithm can be used for construction of large QC-LDPC codes as well.

## V. CONCLUSION

In this paper, a simulated annealing algorithm is proposed to generate QC-LDPC codes that avoids small elementary trapping sets. Among the modern algorithms, the simulated annealing algorithm generates the best QC-LDPC codes in terms of the ETS distribution. Some dominant elementary trapings are completely removed from the Tanner graph by the simulated annealing algorithm. Moreover, Monte Carlo simulations clearly show that proposed QC-LDPC codes are superior to existing designs for QC-LDPC codes.

## REFERENCES

- [1] J. Thorpe, "Low-Density Parity-Check (LDPC) Codes Constructed from Protographs," *Interplanetary Network Progress Report*, vol. 154, pp. 1–7, Aug. 2003.
- [2] I. Eizemendi, M. Velez, D. Gmez-Barquero, J. Morgade, V. Baena-Lecuyer, M. Slimani, and J. Zoellner, "DVB-T2: The second generation of terrestrial digital video broadcasting system," *IEEE Transactions on Broadcasting*, vol. 60, no. 2, pp. 258–271, June 2014.
- [3] T. Richardson and S. Kudekar, "Design of low-density parity check codes for 5G new radio," *IEEE Communications Magazine*, vol. 56, no. 3, pp. 28–34, March 2018.
- [4] T. Richardson, "Error floors of LDPC codes," in *Proceedings of the Annual Allerton Conference on Communication Control and Computing*, 2003, vol. 41, pp. 1426–1435.
- [5] X.-Y. Hu, E. Eleftheriou, and D.M. Arnold, "Regular and irregular progressive edge-growth Tanner graphs," *IEEE Transactions on Information Theory*, vol. 51, no. 1, pp. 386–398, 2005.
- [6] A. Tasdighi, A. H. Banihashemi, and M. Sadeghi, "Efficient search of girth-optimal QC-LDPC codes," *IEEE Transactions on Information Theory*, vol. 62, no. 4, pp. 1552–1564, April 2016.
- [7] M. Diouf, D. Declercq, M. Fossorier, S. Ouya, and B. Vasi, "Improved PEG construction of large girth QC-LDPC codes," in *2016 9th International Symposium on Turbo Codes and Iterative Information Processing (ISTC)*, Sep. 2016, pp. 146–150.
- [8] M. Diouf, D. Declercq, S. Ouya, and B. Vasic, "A PEG-like LDPC code design avoiding short trapping sets," in *2015 IEEE International Symposium on Information Theory (ISIT)*, June 2015, pp. 1079–1083.
- [9] X. Tao, Y. Li, Y. Liu, and Z. Hu, "On the construction of LDPC codes free of small trapping sets by controlling cycles," *IEEE Communications Letters*, vol. 22, no. 1, pp. 9–12, Jan 2018.
- [10] S. Naseri and A. H. Banihashemi, "Construction of girth-8 QC-LDPC codes free of small trapping sets," *IEEE Communications Letters*, pp. 1–1, 2019.

- [11] M. Karimi and A. H. Banihashemi, "On characterization of elementary trapping sets of variable-regular LDPC codes," *IEEE Transactions on Information Theory*, vol. 60, no. 9, pp. 5188–5203, Sept 2014.
- [12] Y. Hashemi and A. H. Banihashemi, "On characterization and efficient exhaustive search of elementary trapping sets of variable-regular LDPC codes," *IEEE Communications Letters*, vol. 19, no. 3, pp. 323–326, March 2015.
- [13] Bashirreza Karimi and Amir H. Banihashemi, "Construction of QC-LDPC codes with low error floor by efficient systematic search and elimination of trapping sets," *CoRR*, vol. abs/1902.07332, 2019.
- [14] R.M. Tanner, D. Sridhara, A. Sridharan, T.E. Fuja, and D.J. Costello Jr, "LDPC block and convolutional codes based on circulant matrices," *IEEE Transactions on Information Theory*, vol. 50, no. 12, pp. 2966–2984, 2004.