Covariance Evolution for Spatially “Mt. Fuji” Coupled LDPC Codes

Yuta Nakahara  
Center for Data Science  
Waseda University  
27 Waseda-cho, Shinjuku-ku, Tokyo, 162-0042 Japan  
E-mail: yuta.nakahara@aoni.waseda.jp

Toshiyasu Matsushima  
Department of Pure and Applied Mathematics  
Waseda University  
3-4-1 Okubo, Shinjuku-ku, Tokyo, 169-8555 Japan  
E-mail: toshimat@waseda.jp

Abstract—A spatially “Mt. Fuji” coupled low-density parity check ensemble is a modified version of the original spatially coupled low-density parity check ensemble. It is known that it has almost the same decoding error probability as and requires less number of iterations than the original ensemble in the waterfall region if we appropriately choose a parameter $\alpha$. In previous studies, a system of differential equations called covariance evolution is derived to analyze the waterfall performance of the original ensemble. In this paper, we modify it for the spatially “Mt. Fuji” coupled low-density parity check ensemble. Then, we analyze the waterfall performance.

Index Terms—spatially coupled codes, covariance evolution, finite-length code performance

I. INTRODUCTION

A spatially coupled (SC) low-density parity check (LDPC) ensemble is constructed as a set of random bipartite graphs like a chain of $(2L+1)$ block LDPC codes whose code lengths are $M$. Then, the code length of the SC-LDPC ensemble is $N = (2L+1)M$. If $M$ and $L$ are sufficiently large, the SC-LDPC ensemble has many desirable properties. In particular, the belief propagation (BP) threshold $\epsilon_{BP}$ of the SC-LDPC ensemble coincides with the maximum a posteriori (MAP) threshold of the underlying block LDPC ensemble for sufficiently large $M$. Moreover, the design rate of the SC-LDPC ensemble converges to the design rate of the underlying block LDPC ensemble for sufficiently large $L$ with $O(L^{-1})$. Note that the design rate is independent of $M$.

However, some problems occur when the code length $N = (2L+1)M$ is fixed to a finite value. In order to increase the design rate, we have to increase $L$ and decrease $M$. If $L$ is too large, the average number of iterations of BP decoding increases in the waterfall region. If $M$ is too small, the decoding error probability increases in the error floor region.

For this problems, a generalized SC-LDPC ensemble is proposed, which is called spatially “Mt. Fuji” coupled (SFC) LDPC ensemble. In the SFC-LDPC ensemble, code lengths of the underlying LDPC codes are different from each other. As the position of the underlying code gets close to the middle of the chain, its code length increases exponentially. The increasing rate is expressed by a parameter $\alpha \geq 1$. Therefore, the design rate of the SFC-LDPC ensemble converges to the design rate of the underlying LDPC ensemble with $O(\alpha^{-L})$ as $L \to \infty$. In the following, we assume that the design rate and the code length of the SFC-LDPC ensemble and those of the SC-LDPC ensemble are equal to each other and $\alpha > 1$. Then, $L$ of the SFC-LDPC ensemble becomes smaller and $M$ of the SFC-LDPC ensemble becomes larger than those of the SC-LDPC codes.

In the error floor region, the decoding error probability of the SFC-LDPC ensemble is lower than that of the SC-LDPC ensemble. That is first observed experimentally in [3]. Later, that is theoretically explained by a weight distribution analysis for the SFC-LDPC ensemble in [4].

The average number of iterations of the SFC-LDPC ensemble in the waterfall region is lower than that of the SC-LDPC ensemble. That is theoretically expected by an observation of the “decoding wave” given by the density evolution and confirmed by numerical experiments in [3].

The decoding error probability of the SFC-LDPC ensemble in the waterfall region is lower than or almost equal to that of the SC-LDPC ensemble if $\alpha$ is appropriately tuned. If $\alpha$ is too large, the decoding error probability becomes larger. This phenomenon is observed experimentally in [3]. It is guessed in [3] as an effect of decrease of the BP threshold $\epsilon_{BP}$, which is regarded as an asymptotic indicator of the waterfall performance. However, the decoding error probability under a finite code length has not been theoretically analyzed yet.

For the original SC-LDPC ensemble, the finite-length decoding error probability in the waterfall region is analyzed by two systems of differential equations called expected graph evolution (EGE) and covariance evolution (CE). The EGE for the SFC-LDPC ensemble has been proposed in [5]. In this paper, we derive the CE for the SFC-LDPC ensemble and combine them. Then, we explain the above phenomenon more theoretically and directly than in previous studies.

II. PRELIMINARY

At first, we describe some notations. Let $\mathbb{N}$, $\mathbb{Z}$, and $\mathbb{Q}$ denote the set of natural numbers, integers, and rational numbers, respectively. For any integers $i$ and $j$, let $[i,j]$ denote the set of $\{i, i+1, \ldots, j\}$. Let $\lceil\cdot\rceil$ denote the ceiling function.
A. \((d_v, d_c, L, \alpha)\) spatially “Mt. Fuji” coupled LDPC ensemble

In this section, we describe the spatial coupling of LDPC codes with increasing code length. The constructed ensemble is called a spatially “Mt. Fuji” coupled (SFC) LDPC ensemble. In the SFC-LDPC ensemble, the code length of LDPC code at position \(i \in [-L, L]\) is \([\alpha L^{-i}] M\) \((\alpha \in \mathbb{Q}, \alpha \geq 1, M \in \mathbb{N})\). This is in contrast to the usual SC-LDPC ensemble, where the code length of every LDPC code is \(M\). Although we can define an ensemble like \([1]\) with smoothing parameter \(w\), we describe only the definition of an ensemble like \([6]\), which is suitable for finite-length analysis.

A. \((d_v, d_c, L, \alpha)\) SFC-LDPC ensemble is defined as a set of random bipartite graphs which are constructed by the following 4 steps.

1) Set variable nodes

At position \(i \in [-L, L] , L \in \mathbb{N} , [\alpha L^{-i}] M\) variable nodes of degree \(d_v\) \(\in \mathbb{N}\) are set. At position \(i \in [-L - d_v + 1, -L - 1] \cup [L + 1, L + d_v - 1] , [\alpha L^{-i}] M\) dummy nodes of degree \(d_v\) are set. The dummy nodes are shortened at the last step.

2) Extend edges deterministically

The \(j\)th \((j \in [0, d_v - 1])\) edge of the variable (or dummy) node at position \(i \in [-L - d_v + 1, L + d_v - 1]\) is extended to the position \(i + j\). Therefore, each variable (or dummy) node extends just one edge to each of the next \(d_v\) positions deterministically.

3) Set check nodes

Because of the above steps, \(\sum_{j=0}^{d_v-1} [\alpha L^{-i-j}] M\) edges come from variable nodes at positions \(i, i-1, \ldots, i-d_v+1\) to the position \(i\) of the check node side. Then, we set \([\frac{1}{d_v} \sum_{j=0}^{d_v-1} [\alpha L^{-i-j}] M\] check nodes at position \(i \in [-L, L + d_v - 1] (d_v \in \mathbb{N})\). In order to equalize the number of edges, only one check node has degree

\[
\rho_i = \sum_{j=0}^{d_v-1} [\alpha L^{-i-j}] M
\]

and the others have degree \(d_v\).

4) Connect edges probabilistically

At each position, the edges of the check nodes are connected to the variable or dummy nodes according to a random permutation of \(\sum_{j=0}^{d_v-1} [\alpha L^{-i-j}] M\) letters. Finally, dummy nodes are shortened.

Remark 1: Under a fixed code length and a design rate, as \(\alpha\) increases, \(M\) increases, and \(L\) decreases because of the above definition \([3]\).

B. Probabilistic properties of the ensemble

If \(M\) is sufficiently large and \(\alpha L^{-i} M\) and both \(\frac{1}{d_v} \sum_{j=0}^{d_v-1} \alpha L^{-i-j} M\) are natural number, the following lemma holds, where sampling without replacement of edges are approximated by sampling with replacement.

**Lemma 1** (Probabilistic property of \((d_v, d_c, L, \alpha)\) SFC-LDPC ensemble):

1) The \(j\)th \((j \in [0, d_v - 1])\) edge of a variable node at position \(i \in [-L, L]\) is connected to a check node at position \(i + j\) with probability 1.

2) An edge of a check node at position \(i \in [-L, L + d_v - 1]\) is connected to a variable or dummy node at position \(i - j, j \in [0, d_v - 1]\) with probability \(\alpha L^{-i-j} / \sum_{k=0}^{d_v-1} \alpha L^{-i-k}\).

3) An edge of a check node at position \(i \in [-L, L + d_v - 1]\) is connected to a variable (not dummy) node in the range of positions \([i - d_v + 1, i]\) with probability \((1 - (1 - s_{i,\alpha})^{d_v})\).

4) At least one edge of a check node at position \(i \in [-L, L + d_v - 1]\) is connected to a variable (not dummy) node in the range of positions \([i - d_v + 1, i]\) with probability \((1 - \rho_{m,i,\alpha})^{d_v}\).

5) A check node at position \(i\) has a degree \(m\) with probability

\[
\rho_{m,i,\alpha} = \begin{cases} 
1, & i \in [-L + d_v - 1, L], m = d_v, \\
0, & i \in [-L + d_v - 1, L], m < d_v, \\
\left(\frac{d_v}{d_v - m}\right)^m (1 - s_{i,\alpha})^{d_v - m}, & i \in [-L, L + d_v - 2] \cup [L + 1, L + d_v - 1].
\end{cases}
\]

III. COVARIANCE EVOLUTION FOR THE SFC-LDPC ENSEMBLE

In this section, we describe the CE for the SFC-LDPC ensemble in order to analyze the decoding error probability in the waterfall region. In the following, we assume that codewords are transmitted through the binary erasure channel with channel erasure probability \(\epsilon = \text{BEC}(\epsilon)\). In addition, the peeling decoder \([7]\) is assumed in the analysis. It has the same decoding error probability as the BP decoder in a sufficiently large number of iterations. Let \(t\) denote the iteration number of the peeling decoder. Let \(V_u(t)\) denote the number of variable nodes at position \(u \in [-L, L + d_v - 1]\) in the residual graph. Let \(R_{j,u}(t)\) denote the number of edges connected to the check nodes of degree \(j \in [1, d_v]\) at the position \(u \in [-L, L + d_v - 1]\) in the residual graph. And their normalized versions are defined by \(\tau = t/M, v_u(\tau) = V_u(\tau M)/M, \) and \(r_{j,u}(\tau) = R_{j,u}(\tau M)/M\).

As \(M \to \infty\), the expected behavior \(\hat{v}_u(\tau) = \mathbb{E}[v_u(\tau)]\) and \(\hat{r}_{j,u}(\tau) = \mathbb{E}[r_{j,u}(\tau)]\) of the peeling decoder for the SC-LDPC ensemble over the BEC(\(\epsilon\)) is known to satisfy a system of differential equations called expected graph evolution (EGE) \([6]\), where the expectation is taken over the ensemble, channel outputs, and the random choice of a degree 1 check node made by the peeling decoder. In addition, let \(\delta_{j,u}^{\tau} = \text{CoVar}[\hat{r}_{j,u}(\tau), r_{j,u}(\tau)]/\mathbb{E}[\hat{r}_{j,u}(\tau)]\) and \(\delta_{j,u}^{\tau} = \text{Var}(\hat{r}_{j,u}(\tau))/\mathbb{E}[\hat{r}_{j,u}(\tau)]\). \(\delta_{j,u}^{\tau}\) is known to satisfy a system of differential equations called covariance evolution (CE) \([6]\) as \(M \to \infty\). Moreover, \(r_{j,u}(\tau)\) is Gaussian distributed with mean \(\hat{r}_{j,u}(\tau)\) and variance \(\delta_{j,u}^{\tau} / \mathbb{E}[\hat{r}_{j,u}(\tau)]\) for a sufficiently large \(M\).
Since the decoding rule for the SFC-LDPC ensemble is the same as that for the SC-LDPC ensemble, the difference between the SFC-LDPC ensemble and the SC-LDPC ensemble appears in the initial conditions of the EGE and the CE. The initial condition of the EGE for the SFC-LDPC ensemble has been proposed in [5]. We reproduce it in the Appendix

In this paper, we derive the initial conditions of the CE has been proposed in [5]. We reproduce it in the Appendix

A. In this paper, we derive the initial conditions of the CE and the CE. The initialization of those for the SC-LDPC ensemble, we describe

\[ \rho_{\alpha}^L = \begin{cases} 1, & u \in [-L, L + d_u - 1, L], m = d_c, \\ 0, & u \in [-L + d_u - 1, L], m < d_c, \\ \left( \frac{d_c - 1}{m - 1} \right) (s_{u, \alpha})^{m - 1} \left( 1 - s_{u, \alpha} \right)^{d_c - m}, & u \in [-L, -L + d_v - 2] \cup [L + 1, L + d_v - 1]. \end{cases} \]

\[ P(d_u = j, d_x = z|\text{no share}) = \begin{cases} \sum_{m=j}^{d_c} \rho_{m,u,\alpha} (m - 1) e^{m - 1} (1 - \epsilon)^{m - j}, & \text{for } \epsilon \leq 0.4, \\ (m - 1) e^{m - 1} (1 - \epsilon)^{m - j}, & \text{for } \epsilon > 0.4. \end{cases} \]

Remark 2: If \( \alpha = 1 \), the above initial condition coincides with that for the SC-LDPC ensemble in [6]. Therefore, this is a natural generalization of it.

IV. PREDICTION OF THE DECODING ERROR PROBABILITY

OF THE SFC-LDPC ENSEMBLE

In this section, we combine the solution of the EGE for SFC-LDPC ensemble [5] with the solution of the CE for SFC-LDPC ensemble derived in the preceding section in order to predict the finite-length decoding error probability of the SFC-LDPC ensemble.

We reproduce the average number of the degree 1 check nodes \( \hat{r}_1(\tau) = \sum_{u=-L}^{L} \hat{r}_{1,u}(\tau) \) calculated from the solution of the EGE for the SFC-LDPC ensemble in Fig. 1 which is derived numerically with the classical Runge-Kutta method in [5]. Figure 2 shows the variance of the number of degree 1 check nodes \( \delta_1(\tau) = \sum_{u=-L}^{L} \sum_{x=-L}^{L} \delta_{1,x}^L(\tau) \) calculated from the solution of the CE for the SFC-LDPC ensemble, which is derived numerically with the Euler’s method. \( \hat{r}_1(\tau) \) and \( \delta_1(\tau) \) have a local minimum, and the smaller \( \epsilon \) is, the “sharper” \( \hat{r}_1(\tau) \) is too.

This is a special feature of the SFC-LDPC ensemble because the previous SC-LDPC ensemble does not have such a local minimum but a flat part called a critical phase. Therefore, they had to regard \( r_1(\tau) \) of the previous SC-LDPC ensemble as an Ornstein-Uhlenbeck process to approximate the block error probability. However, we are able to approximate the block error probability of the SFC-LDPC ensemble by the
probability that the error event occurs on the local minimum like the case of block LDPC codes [8]. Note that the error event occurs when \( r_1(\tau) = 0 \) before all nodes are removed.

Let \( \tau^* \) denote the time when \( r_1(\tau) \) is the local minimum point. As shown in Fig. 1, \( r_1(\tau^*) \) looks almost proportional to \( \epsilon_{BP} - \epsilon \), where BP thresholds of the (3,6,20,1.1) and (3,6,25,1.05) SFC-LDPC ensemble are 0.4703 and 0.4785, respectively. On the other hand, \( \delta_1(\tau^*) \) is almost constant for \( \epsilon \). Then, we approximate \( r_1(\tau^*) \) by \( \gamma(\epsilon_{BP} - \epsilon) \), and approximate the ensemble average block error probability by

\[
Q \left( \frac{\gamma(\epsilon_{BP} - \epsilon)}{\sqrt{n/\delta_1(\tau^*)} / M} \right),
\]

where \( Q(\cdot) \) denotes the Q-function, and \( \gamma \) is calculated by \( \gamma = \epsilon_{BP} - \epsilon \) for each \( L \) and \( \alpha \). Therefore, we consider that the BP threshold \( \epsilon_{BP} \) affects the “position” of the waterfall and the coefficient \( \gamma/\sqrt{\delta_1(\tau^*)} \) affects the “steepness” of the waterfall. Table I shows \( \epsilon_{BP} \), \( \gamma \), \( \delta_1(\tau^*) \), and \( \gamma/\sqrt{\delta_1(\tau^*)} \) of \( (3,6,20,\alpha) \) SFC-LDPC ensemble for several \( \alpha \). Then, we expect that the larger \( \alpha \) is, the steeper the waterfall is and the more left-shifted. Note that \( M \) also increases as \( \alpha \) increases, as described in Remark 1.

### V. Experiments

#### A. Experiment conditions

The parameters of the ensembles used in the experiments are shown in Table II. We generate 1100 codes and 1000 codewords from each code for A1–A4 with \( \epsilon \leq 0.440 \), B1 with \( \epsilon \leq 0.450 \), and C1 with \( \epsilon \leq 0.455 \), and we generate 100 codes and 1000 codewords from each code for A1–A4 with \( \epsilon \geq 0.445 \), B1 with \( \epsilon \geq 0.455 \), and C1 with \( \epsilon \geq 0.460 \). The rates shown in Table II are the average rate of those generated codes. The decoder is the BP decoder with no limitation of the number of iterations, which has the same decoding error probability as that of the peeling decoder. Note that we remove the small cycles of Tanner graphs, whose lengths are lower than or equal to 6, in order to observe the block error probability in the waterfall region more precisely.

#### B. Difference in \( M \)

Figure 3 shows that the simulated block error probability curves and the estimated curves for several \( M \) are the same decoding error probability as that of the peeling decoder. Note that we remove the small cycles of Tanner graphs, whose lengths are lower than or equal to 6, in order to observe the block error probability in the waterfall region more precisely.

#### C. Difference in \( L \)

Figure 4 shows that the simulated block error probability curves and the estimated curves for several \( L \) are the same decoding error probability as that of the peeling decoder. Note that we remove the small cycles of Tanner graphs, whose lengths are lower than or equal to 6, in order to observe the block error probability in the waterfall region more precisely.

#### D. Difference in \( \alpha \)

Figure 5 shows that block error probability curves and estimated curves for several \( \alpha \) are the same decoding error probability as that of the peeling decoder. Note that we remove the small cycles of Tanner graphs, whose lengths are lower than or equal to 6, in order to observe the block error probability in the waterfall region more precisely.
TABLE III
THE PARAMETERS OF THE TARGET SC-LDPC ENSEMBLE AND THE CONSTRUCTED SFC-LDPC ENSEMBLE

|         | \(d_{e}\) | \(d_{c}\) | \(L_{c}\) | \(\alpha\) | \(M\) | Code length | Rate   |
|---------|-----------|-----------|-----------|-----------|-----------|-------------|--------|
| Target  | 3         | 6         | 25        | 1.00      | 250       | 12,750      | 0.482  |
| SFC     | 3         | 6         | 12        | 1.11      | 260       | 12,734      | 0.483  |

the larger \(\alpha\) is, the sharper the graph of \(\hat{r}_1(\tau)\) around the local minimum is, as shown in Fig. 1.

E. Code construction

From the preceding analysis, we expect that the decoding error probability in the waterfall region is left-shifted and becomes steeper as \(\alpha\) increases. In addition, it has already been observed in [3] that the average number of iterations of the SFC-LDPC ensemble in the waterfall region is less than that of the SC-LDPC ensemble. Then, tuning \(\alpha\) appropriately, we can construct an SFC-LDPC ensemble with the following 3 properties. 1. It has the same rate and code length as those of the target SC-LDPC ensemble. 2. It has a lower decoding error probability than the target ensemble under the condition \(\epsilon < \epsilon^*\), where \(\epsilon^*\) is a target channel erasure probability. 3. It has a lower average number of iterations than that of the target ensemble.

Actually, we construct an SFC-LDPC ensemble with the above properties. The parameters of the target SC-LDPC ensemble and the constructed SFC-LDPC ensemble are in Table III. The small cycles of Tanner graphs, whose lengths are lower than or equal to 6 are removed. Target channel erasure probability is \(\epsilon^* = 0.44\). Figure 5 shows the decoding error probability of those ensembles derived by Monte Carlo simulation. When \(\epsilon = 0.42\), 11000 codes and 100 codewords from each code are generated. When \(\epsilon > 0.42\), 1000 codes and 100 codewords from each code are generated. Figure 6 shows the average number of iterations of them. The constructed ensemble has the desired properties.

VI. CONCLUSION

In this paper, we derived the CE for the SFC-LDPC ensemble. We combined its solution with the solution of the EGE for the SFC-LDPC ensemble, which had been derived in [5]. Then, we analyzed the decoding error probability of the SFC-LDPC ensemble in the waterfall region. The waterfall became steeper as \(\alpha\) increased. As a result, it mitigated the decrease of the BP threshold.

APPENDIX A

For \(u \in [-L, L + d_{c} - 1], j \in [1, d_{c}]\), the initial conditions of the EGE for SFC-LDPC ensemble are as follows.

\[
\bar{r}_{j,u}(0) = j \frac{1}{d_{c}} \left( \sum_{k=0}^{d_{c}-1} \alpha^{L-|u-k|} \right) \sum_{m \geq j} \rho_{m,u,\alpha} \binom{m}{j} e^{j(1-\epsilon)^{m-j}},
\]  

where \(\rho_{m,u,\alpha}\) is defined in (2).

\[
\bar{v}_{u}(0) = \begin{cases} \alpha^{L-|u|}, & u \in [-L, L], \\ 0, & \text{otherwise}. \end{cases}
\]  

APPENDIX B

Our proof mainly follows the Appendix C of [6] but more detailed and generalized. In addition to the notation in Section III, let \(\delta_{2,z}^{u,\alpha}(\tau)\) also denote \(\text{CoVar}[v_{u}(tM), R_{z,p}(tM)]/M = \text{CoVar}[v_{u}(\tau), v_{z,\delta}(\tau)]M\) for \(j = d_{c} + 1, z \leq d_{c}\) and \(\text{CoVar}[v_{u}(tM), V_{z}(tM)]/M = \text{CoVar}[v_{u}(\tau), v_{z}(\tau)]M\) for \(j = z = d_{c} + 1\). Initial conditions of the covariance evolution are as follows. Let \(p_{j,u,\alpha}\) denote the probability that a randomly chosen check node at position \(u\) has the degree \(j\) after the peeling decoder initialization. It is given by

\[
p_{j,u,\alpha} = \sum_{m=j}^{d_{c}} \rho_{m,u,\alpha} \binom{m}{j} e^{j(1-\epsilon)^{m-j}}.
\]
Initial conditions of the covariance evolution for the SFC-LDPC ensemble are divided into the following three cases:

\[
\delta_{z,u}^{l+1}(0) = \begin{cases} 
\text{CoVar}[R_{j,u}(0), R_{z,x}(0)]/M, & j, z \leq d_c, \\
\text{CoVar}[V_u(0), R_{z,x}(0)]/M, & j = d_x + 1, z \leq d_c, \\
\text{CoVar}[V_u(0), V_z(0)]/M, & j = z = d_x + 1 
\end{cases}
\]

When \( j = z = d_c + 1, V_u(0) \) follows a binomial distribution with \( \alpha L^{-|u|} \) trials and probability \( \epsilon \) independently from each other position. Therefore,

\[
\delta_{z,u}^{d+1}(0) = \text{CoVar}[V_u(0), V_z(0)]/M = \text{Var}[V_u(0)]/M = \alpha L^{-|u|}\epsilon(1 - \epsilon), \quad u = x, \quad u \neq x
\]

When \( j, z \leq d_c \), we divide the cases as follows.

\[
\begin{align*}
\delta_{j,u}^{l}(0) & = \text{Var}[R_{j,u}(0)]/M \\
& = j^2 \frac{1}{d_c} \left( \sum_{k=0}^{d_c - 1} \alpha L^{-|u-k|} \right) p_{j,u,\alpha}(1 - p_{j,u,\alpha})
\end{align*}
\]

and for \( u = x \) and \( j \neq z \),

\[
\delta_{z,u}^{l}(0) = \text{CoVar}[R_{j,u}(0), R_{z,u}(0)]/M = -j \frac{1}{d_c} \left( \sum_{k=0}^{d_c - 1} \alpha L^{-|u-k|} \right) p_{j,u,\alpha} p_{z,u,\alpha}
\]

For \( u \neq x \) and \( |u - x| \geq d_v \), any check node at position \( u \) and any check node at position \( x \) cannot be connected to each other by one variable node, and they are independent through the peeling decoder initialization. Therefore,

\[
\delta_{x,u}^{l}(0) = \text{CoVar}[R_{j,u}(0), R_{z,x}(0)]/M = 0.
\]

For \( u \neq x \) and \( |u - x| < d_v \), we assume \( u > x \) without loss of generality. In this case, we have to consider the effect from a check node at position \( u \) and a check node at position \( x \) which share at least one variable node before the peeling node shortening and the initialization of the peeling decoder. Let check\(_u\) and check\(_x\) denote a pair of check nodes selected at random from positions \( u \) and \( x \), respectively. There are \( d_v - |u - x| \) positions, from \( x - d_v + 1 \) to \( u \), in which any variable node is connected with one edge to a check node at position \( u \) and with one edge to a check node at position \( x \). check\(_u\) has \( a \) edges connected to variable nodes at positions \( [x - d_v + 1, u] \), and the number \( a \) is according to a binomial distribution with \( d_c \) trials with probability

\[
\sum_{j=0}^{u} \frac{\alpha L^{-|u-j|}}{\sum_{j=0}^{u} \alpha L^{-|u-j|}}.
\]

check\(_x\) has \( b \) edges in the same manner as check\(_u\). Note that \( a \) and \( b \) are independent random variables.

For a given pair \((a, b)\), the probability that check\(_u\) and check\(_x\) share at least one variable node at positions \( [x - d_v + 1, u] \) is approximated as follows, where the sampling without replacement is approximated by the sampling with replacement.

\[
1 - \frac{\left( \sum_{k=x-d_v+1}^{u} \frac{\alpha L^{-|u-j|}}{\sum_{j=0}^{u} \alpha L^{-|u-j|}} \right)^b ab}{\sum_{k=x-d_v+1}^{u} \alpha L^{-|u-j|}} ^ M.
\]

for sufficiently large \( M \) by ignoring the terms \( O(M^{-2}) \). We ignore the case that check\(_u\) and check\(_x\) share two or more variable nodes since the probability of such a case decays by \( O(M^{-2}) \). Then, averaging over all possible pairs \((a, b)\), we can evaluate the probability \( P_S \) that check\(_u\) and check\(_x\) share at least one variable node before dummy node shortening by

\[
P_S = \frac{1}{\sum_{k=x-d_v+1}^{u} \alpha L^{-|u-j|}} \cdot \frac{\sum_{k=0}^{u} \frac{\alpha L^{-|u-j|}}{\sum_{j=0}^{u} \alpha L^{-|u-j|}} \cdot \frac{\sum_{k=x-d_v+1}^{u} \alpha L^{-|u-j|}}{\sum_{j=0}^{u} \alpha L^{-|u-j|}}}{\left( \sum_{j=0}^{d_v} \alpha L^{-|u-j|} \right) \left( \sum_{j=0}^{d_v} \alpha L^{-|u-j|} \right)}.
\]

Because the probability that the shared variable node is not a dummy node is

\[
\sum_{k=0}^{d_v} \frac{\alpha L^{-|u-j|}}{\sum_{j=0}^{u} \alpha L^{-|u-j|}} ^ M,
\]

the probability \( P_S' \) that check\(_u\) and check\(_x\) share at least one variable node after dummy node shortening before the peeling decoder initialization is

\[
P_S' = \frac{\sum_{k=0}^{d_v} \frac{\alpha L^{-|u-j|}}{\sum_{j=0}^{u} \alpha L^{-|u-j|}}}{\sum_{k=0}^{d_v} \frac{\alpha L^{-|u-j|}}{\sum_{j=0}^{u} \alpha L^{-|u-j|}}} \cdot \frac{\sum_{k=x-d_v+1}^{u} \alpha L^{-|u-j|}}{\sum_{j=0}^{d_v} \alpha L^{-|u-j|}} ^ M \left( \sum_{j=0}^{d_v} \alpha L^{-|u-j|} \right) \left( \sum_{j=0}^{d_v} \alpha L^{-|u-j|} \right) \left( \sum_{j=0}^{d_v} \alpha L^{-|u-j|} \right)
\]

Let \( d_x \) and \( d_y \) denote the degree of check\(_u\) and check\(_x\) after the peeling decoder initialization, respectively. Then, the
probability that $P(d_u = j, d_x = z)$ can be expressed as follows.

$$ P(d_u = j, d_x = z) = P(d_u = j, d_x = z | \text{share}) P'_S + P(d_u = j, d_x = z | \text{no share})(1 - P'_S), \quad (25) $$

where $P(d_u = j, d_x = z | \text{share})$ denotes the conditional probability that check$_u$ and check$_u$ share one variable (not dummy) node. It is obtained by

$$ P(d_u = j, d_x = z | \text{share}) = e \left[ \sum_{m=j}^{d_c} \rho'_{m,u,\alpha} \left( \frac{m-1}{j-1} \right) \left( 1 - \epsilon \right)^{m-j} \right] \times \left( \sum_{m=z+1}^{d_c} \rho'_{m,x,\alpha} \left( \frac{m-1}{z-1} \right) \left( 1 - \epsilon \right)^{m-z-1} \right) \times (1 - \epsilon) \left[ \sum_{m=j+1}^{d_c} \rho'_{m,u,\alpha} \left( \frac{m-1}{j} \right) \left( 1 - \epsilon \right)^{m-j-1} \right], \quad (26) $$

where

$$ \rho'_{m,u,\alpha} = \begin{cases} 
1, & u \in [-L + d_u - 1, L], m = d_c, \\
0, & u \in [-L + d_u - 1, L], m < d_c, \\
\left( \frac{d_c}{m-1} \right) (s_{u,\alpha})^{m-1} (1 - s_{u,\alpha})^{d_c-m}, & u \in [-L, -L + d_u - 2] \cup [L + 1, L + d_u - 1]. 
\end{cases} \quad (27) $$

The first term of $P(d_u = j, d_x = z | \text{share})$ represents the probability that the shared variable node is erased, and the second term represents the probability that the shared variable node is not erased. In addition, the following holds directly.

$$ P(d_u = j, d_x = z | \text{no share}) = p_{j,u,\alpha} p_{z,x,\alpha}. \quad (28) $$

Then, $\text{CoVar}[R_{j,u}(0), R_{z,x}(0)]$ is obtained by the following calculation.

$$ \text{CoVar}[R_{j,u}(0), R_{z,x}(0)] = E[R_{j,u}(0) R_{z,x}(0)] - E[R_{j,u}(0)] E[R_{z,x}(0)] \quad (29) $$

$$ = j \left( \frac{M}{d_c} \sum_{k=0}^{d_c-1} \alpha L^{-|u-k|} \right) \cdot z \left( \frac{M}{d_c} \sum_{k=0}^{d_c-1} \alpha L^{-|x-k|} \right) \cdot P(d_u = j, d_x = z) $$

$$ = j \left( \frac{M}{d_c} \sum_{k=0}^{d_c-1} \alpha L^{-|u-k|} \right) p_{j,u,\alpha} $$

$$ \cdot z \left( \frac{M}{d_c} \sum_{k=0}^{d_c-1} \alpha L^{-|x-k|} \right) p_{z,x,\alpha} \quad (30) $$

$$ = jz \left( \frac{M}{d_c} \sum_{k=0}^{d_c-1} \alpha L^{-|u-k|} \right) \left( \frac{M}{d_c} \sum_{k=0}^{d_c-1} \alpha L^{-|x-k|} \right) $$

$$ \cdot \left\{ P'_S \left( P(d_u = j, d_x = z | \text{share}) - P(d_u = j, d_x = z | \text{no share}) \right) + p_{j,u,\alpha} p_{z,x,\alpha} \right\} \quad (31) $$

$$ = jz \left( \frac{M}{d_c} \sum_{k=0}^{d_c-1} \alpha L^{-|u-k|} \right) \left( \frac{M}{d_c} \sum_{k=0}^{d_c-1} \alpha L^{-|x-k|} \right) $$

$$ \cdot \left\{ p_{j,u,\alpha} p_{z,x,\alpha} \right\} $$

Therefore,

$$ \delta_{z,x}^j(0) = \text{CoVar}[R_{j,u}(0), R_{z,x}(0)] / M $$

$$ = \left\{ \frac{\min(L,u)}{k=\max(-L,x-d_u+1)} \sum_{k=\max(-L,x-d_u+1)}^{\min(L,u)} \alpha L^{-|k|} \right\} \times \left\{ P(d_u = j, d_x = z | \text{share}) - P(d_u = j, d_x = z | \text{no share}) \right\}. \quad (33) $$

When $j = d_c + 1$ and $z \leq d_c$, for $0 < x - u < d_v$, in a similar manner,

$$ \delta_{x,z}^{d_c+1}(0) = \text{CoVar}[V_u(0), R_{z,x}(0)] / M $$

$$ = z \left( \sum_{m=z}^{d_c} \rho'_{m,x,\alpha} \left( \frac{m-1}{z-1} \right) \left( 1 - \epsilon \right)^{m-z-1} \right) \cdot \left( \sum_{m=z}^{d_c} \rho'_{m,x,\alpha} \left( \frac{m}{z} \right) \left( 1 - \epsilon \right)^{m-z} \right). \quad (34) $$

and otherwise,

$$ \delta_{x,z}^{d_c+1}(0) = \text{CoVar}[V_u(0), R_{z,x}(0)] / M = 0. \quad (35) $$

REFERENCES

[1] S. Kudekar, T. J. Richardson, and R. L. Urbanke. “Threshold saturation via spatial coupling: Why convolutional ldpc ensembles perform so well over the bsc,” IEEE Transactions on Information Theory, vol. 57, no. 2, pp. 803–834, Feb 2011.

[2] P. M. Olmos and R. Urbanke. “Scaling behavior of convolutional ldpc ensembles over the bsc,” in 2011 IEEE International Symposium on Information Theory Proceedings, July 2011, pp. 1816–1820.

[3] Y. NAKAHARA, S. SAITO, and T. MATSUSHIMA. “Spatially “mt. fuji” coupled ldpc codes,” IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences, vol. E100.A, no. 12, pp. 2594–2606, 2017.
[4] Y. NAKAHARA and T. MATSUSHIMA, “A note on weight distributions of spatially “mt. fuji” coupled ldpc codes,” *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, vol. E101-A, no. 12, pp. 2194–2198, 2018.

[5] Y. Nakahara and T. Matsushima, “Expected graph evolution for spatially “mt. fuji” coupled ldpc codes,” in *2018 International Symposium on Information Theory and Its Applications (ISITA)*, Oct 2018, p. 511.

[6] P. M. Olmos and R. L. Urbanke, “A scaling law to predict the finite-length performance of spatially-coupled ldpc codes,” *IEEE Transactions on Information Theory*, vol. 61, no. 6, pp. 3164–3184, June 2015.

[7] M. G. Luby, M. Mitzenmacher, M. A. Shokrollahi, D. A. Spielman, and V. Stemann, “Practical loss-resilient codes,” in *Proceedings of the Twenty-ninth Annual ACM Symposium on Theory of Computing*, ser. STOC ’97. New York, NY, USA: ACM, 1997, pp. 150–159. [Online]. Available: [http://doi.acm.org/10.1145/258533.258557](http://doi.acm.org/10.1145/258533.258557)

[8] A. Amraoui, “Asymptotic and finite-length optimization of ldpc codes,” EPFL, Tech. Rep., 2006.