


Improved PEG-Based Construction of Analog Fountain Codes

Xue Li  and Qin Huang *

School of Electronic and Information Engineering, Beihang University, Beijing 100191, China; lixue19@buaa.edu.cn

* Correspondence: qhuang.smash@gmail.com; Tel.: +86-010-8231-4663

Abstract: This paper proposes an improved *progressive edge-growth* (PEG) construction of analog fountain codes (AFCs). During edge selection, it simultaneously allocates weight coefficients in descending order. Analysis shows that our proposed construction reduces the probability of large weight coefficients involved in harmful short cycles. Simulation results indicate that it has good block error rate (BLER) in short block length regime.

Keywords: analog fountain codes; progressive edge-growth; weight coefficients

1. Introduction

With the emergence of applications such as intelligent transportation, tele-surgery, and industrial automation, three main service categories are defined in the fifth generation (5G) mobile network. Among these, ultra-reliable low latency communication (URLLC) poses the most significant challenges due to its physical layer design [1]. URLLC requires a short end-to-end latency, along with high reliability evaluated by block error rate (BLER). These strict constraints impose stringent requirements for the design of channel coding in short block length regime.

Recently, analog fountain codes (AFCs) [2–4], a type of rateless codes, have shown their great potential to improve the latency and reliability while keeping bit-level granularity. Several impressive studies [5–12] have explored the coding schemes and constructions of AFCs to make them promising for URLLC. In [5], optimal AFC parameters were analyzed and designed to minimize the bit error rate. In [7], a low-complexity weight-adaptive AFC transmission scheme was proposed based on extrinsic information transfer (EXIT) analysis. In [9], Bose–Chaudhuri–Hocquenghem (BCH) precoders were designed for AFCs to improve performance for short packet communication. In [11], joint design of AFCs and quasi-gray constellation mapping modulation was proposed, approaching the benchmark. In [12], a short partitioned transmission strategy based on AFCs was proposed to further enhance the error rate and latency performance.

AFC generator matrices construction involves two steps: edge selection and weight coefficient allocation. In [2], the edges were selected to maximize the minimum variable node (VN) degree of AFCs. Weight coefficients were allocated randomly. In [6], short AFCs (S-AFCs) reduced error floors by processing these two steps according to a reliability measure. In [8], Online AFC selected edges and allocated weight coefficients according to feedback information, thereby enhancing reliability. Since short cycles deteriorate the performance of the BP decoding, the PEG algorithm [13] was proposed to reduce short cycles of low-density parity-check (LDPC) codes. In [10], an efficient coefficient progressive edge-growth (WC-PEG) algorithm avoided short cycles to obtain good performance. It selected weight coefficients in the same number of times in each column of generator matrix to avoid reducing the rank of generator matrix.

In this paper, we show that large weight coefficients have high mutual information; thus, they contribute most during the BP decoding. Therefore, it is important to avoid them in short cycles, which may severely deteriorate the decoding performance. Note that PEG



Citation: Li, X.; Huang, Q. Improved PEG-Based Construction of Analog Fountain Codes. *Entropy* **2024**, *26*, 841. <https://doi.org/10.3390/e26100841>

Academic Editor: Jun Chen

Received: 30 August 2024

Revised: 29 September 2024

Accepted: 3 October 2024

Published: 5 October 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

maximizes the cycle length, and the earlier selected edges are less likely to be involved in short cycles. We propose to add edges and allocate their weight coefficients in descending order simultaneously. To be specific, we select edges with the smallest reliability measure, and allocate larger weight coefficients to earlier selected edges. Our analysis shows that the proposed construction effectively decreases the probability that larger weight coefficients are involved in short cycles. The simulation results show that it has higher reliability over the existing constructions in short block length regime.

This paper is organized as follows. Section 2 presents the necessary background. Section 3 discusses our proposed construction as well as the simulation results. Section 4 concludes the paper.

2. Analog Fountain Codes and the PEG Algorithm

In this section, encoding and decoding processes of AFCs are briefly introduced. Then, PEG applied to AFC construction is reviewed.

2.1. Preliminaries of AFC

Consider an information sequence \mathbf{b}' comprising k information bits $b'_i \in \{0, 1\}$, where $i = 0, 1, \dots, k - 1$. \mathbf{b}' is binary phase shift keying (BPSK) modulated, resulting in $\mathbf{b} = (b_0, b_1, \dots, b_{k-1})$, where $b_i \in \{-1, 1\}$. Each AFC codeword $\mathbf{c} = \{c_0, c_1, \dots, c_{n-1}\}$ is then generated as

$$\mathbf{c} = \mathbf{G}\mathbf{b}^T. \tag{1}$$

Here, \mathbf{G} is the generator matrix:

$$\mathbf{G} = \begin{bmatrix} g_{0,0} & g_{0,1} & \cdots & g_{0,k-1} \\ g_{1,0} & g_{1,1} & \cdots & g_{1,k-1} \\ \vdots & \vdots & \ddots & \vdots \\ g_{n-1,0} & g_{n-1,1} & \cdots & g_{n-1,k-1} \end{bmatrix} \tag{2}$$

and \mathbf{b}^T is the transpose of \mathbf{b} . The corresponding Tanner graph is shown in Figure 1.

The degree d of AFC is defined as the number of nonzero elements in each row of \mathbf{G} . The d nonzero elements correspond to a predefined weight set $W = \{w_0, w_1, \dots, w_{d-1}\}$, where $w \in W$ is called the weight coefficient. Without loss of generality, we set $w_0 > w_1 > \dots > w_{d-1} > 0$.

After being transmitted through an additive white Gaussian noise (AWGN) channel, $\mathbf{r} = \mathbf{c} + \mathbf{n}$ is received, where \mathbf{n} is the noise vector with zero mean and variance σ^2 .

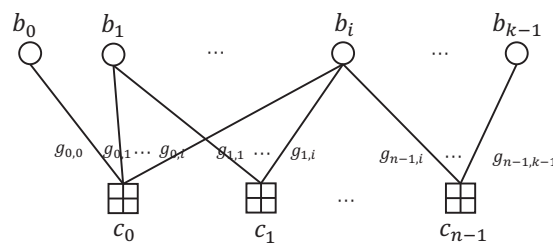


Figure 1. The Tanner graph of an AFC.

At the receiver side, a simplified compressive sensing belief propagation (CS-BP) decoder [14] is employed. The CS-BP decoder is a variant of the BP decoding which can be used in compressive sensing and AFC. The decoding process is briefly reviewed as follows. Let $\mu_{v \rightarrow c}(b_i)$ and $\mu_{c \rightarrow v}(b_i)$ denote the message transmitted from the VNs b_i to check nodes

(CNs) c_j and vice versa, respectively. After a fixed number of iterations of $\mu_{v \rightarrow c(b_i)}$ and $\mu_{c \rightarrow v(b_i)}$, the marginal distribution $f(b_i)$ is obtained by

$$f(b_i) = \prod_{j \in \mathcal{J}_i} \mu_{u \rightarrow v(b_i)}, \tag{3}$$

where \mathcal{J}_i denotes the set of CNs connected to b_i . Then, the decoder outputs the hard decision based on $f(b_i)$. More details of the CS-BP decoder can be reached in [14].

2.2. The PEG Algorithm

Short cycles on Tanner graphs may deteriorate the decoding performance of the BP. Thus, the PEG [13] maximizes the cycle length in the Tanner graph by progressively establishing edges between VNs and CNs in an edge-by-edge manner. VN degree is defined as the number of nonzero elements in each column of \mathbf{G} .

Suppose PEG is used to construct an $n \times k$ matrix \mathbf{G} for AFC. For a given c_j , where $0 \leq j < n$, we define its neighbor VNs within a depth l as $N_{c_j}^l$, and the remaining VNs are denoted by $\overline{N}_{c_j}^l$. If the subgraph from c_j is fully expanded in depth l , two cases may happen. The cardinality of $N_{c_j}^l$ may stop increasing but $|N_{c_j}^l| < k$, or $\overline{N}_{c_j}^l \neq \emptyset$ but $\overline{N}_{c_j}^{l+1} = \emptyset$. Let (c_j, b_i) denote the edge connecting c_j and b_i . At last, the newly established edge is (c_j, b_i) , where b_i is picked from the set $\overline{N}_{c_j}^l$ with the lowest VN degree. Note that length- r cycle arises if $\overline{N}_{c_j}^{l+1} = \emptyset$, where $r = 2(l + 1)$.

When PEG is used to construct generator matrices for AFC, both edge selection and weight coefficients allocation should be considered. In [10], if the edge (c_j, b_i) is selected according to PEG, the weight coefficients are suggested to be allocated according to their appearance in the i -th column of \mathbf{G} . Specifically, we calculate the times that each weight coefficient appears among the nonzero elements in the i -th column. The one with the lowest number of times should be allocated in order to balance the weight of columns in \mathbf{G} .

3. Improved PEG-Based Construction

In this section, EXIT charts of different weight coefficients at CNs are shown. To avoid large weight coefficients involved in short cycles, we propose to add edges and allocate their weight coefficients simultaneously during edge selection. Analysis shows that the proposed construction efficiently reduces the probability of larger weight coefficients involved in the edges of short cycles.

3.1. EXIT Charts of Different Weight Coefficients

Different weight coefficients contribute differently in the decoding process at CNs. To illustrate that, EXIT charts for various weight coefficients are presented below.

Let $\mathbf{b} = \mathbf{b}_0 \cup \mathbf{b}_1 \cup \dots \cup \mathbf{b}_{d-1}$ denote the information bits. Without loss of generality, for each CN c_j connected to d VNs $b_{i_0}, b_{i_1}, \dots, b_{i_{d-1}}$, let $g_{i_s, j} = w_s$, where $s \in \{0, 1, \dots, d - 1\}$. Then $b_{i_s} \in \mathbf{b}_s$. In this way, information bits are divided into d parts, where bits in \mathbf{b}_s are connected to CN through an edge allocated with w_s .

Let $I_{A,C}$ and $I_{E_s,C}$ denote the input mutual information and extrinsic mutual information of the s -th weight coefficient for CN, respectively. For each \mathbf{b}_s , the mutual information between \mathbf{b}_s and its corresponding log-likelihood ratio (LLR) value L_s computed at the CNs is represented as

$$I_{E_s,C} = \sum_{\mathbf{b}_s \in \{-1,1\}} \int_{-\infty}^{+\infty} p_E(L_s|\mathbf{b}_s) \log_2 \frac{p_E(L_s|\mathbf{b}_s)}{p_E(\mathbf{b}_s)} dL_s. \tag{4}$$

Here, p_E represents the conditional probability distribution function of L_s . As can be seen in Equation (4), each calculation of $I_{E_s,C}$ is based on bits \mathbf{b}_s and LLR L_s . It means that after one decoding process at the CNs, CNs output extrinsic mutual information $I_{E_s,C}$ to VNs

which are connected through weight coefficients w_s . It is difficult to compute p_E directly because of the convolution operation. Thus, the Monte Carlo methods are used to estimate p_E for AFC [7]. Note that $L \sim \mathcal{N}(0, \sigma_L)$, where σ_L can be generated from $I_{A,C}$ as

$$\sigma_L = J^{-1}(I_{A,C}). \tag{5}$$

The detailed definition of J -function can be reached in [15].

Figure 2 shows the mutual information for four different weight coefficients of AFC, respectively. It can be seen that with the same $I_{A,C}$, the mutual information is larger when weight coefficient is larger. These results indicate that larger weight coefficients contribute higher in the BP decoding.

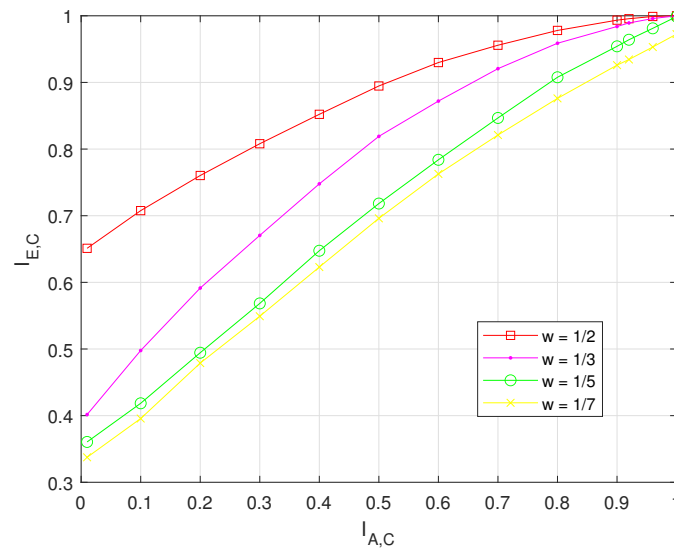


Figure 2. EXIT charts of different weight coefficients, where $d = 4$, $W_s = \{1/2, 1/3, 1/5, 1/7\}$, and signal–noise ratio (SNR) = 18 dB.

3.2. Improved PEG-Based Construction Method

It is well known that short cycles may deteriorate the performance of the BP decoding. Large weight coefficients should not be involved in harmful short cycles due to their high contribution in the BP decoding. AFC construction involves two steps: edge selection and weight coefficient allocation. From these two aspects, we propose to construct generator matrix as follows.

During edge selection, we balance the weight of columns in \mathbf{G} with a reliability measure denoted by $\sum_{p=0}^{i-1} g_{p,q}^2$. The reliability measure represents the sum of squares of weight coefficients for the q -th column of \mathbf{G} before the edges for c_i are established. Apparently, if a column has a relatively low reliability measure, a corresponding edge should be selected to enlarge its reliability measure. Thus, each time the PEG establishes a candidate set of VN for edge selection, the set is further downsized based on the smallest reliability measure set $Q = \{q | \arg \min_q \sum_{p=0}^{i-1} g_{p,q}^2\}$.

With regard to weight coefficient allocation, recall that the PEG maximizes the cycle length, which means that the edges selected earlier are more likely to be in large cycle. In other words, the edges selected later are more likely to be in the short cycle. Thus, they should not be allocated with larger weight coefficients. It forms the weight coefficient allocation strategy that allocating larger weight coefficients to earlier selected edges.

The proposed construction is presented as follows and the details are summarized in Algorithm 1. Given the parameters k, n, d , and W , an edge-selection procedure is initiated. Let $E_{c_i}^j$ denote the j -th edge selected for c_i , where $0 \leq j < d$. The placement of a new edge $E_{c_i}^0$ on the graph considers the reliability measure of each column. Specifically,

$E_{c_i}^0 \leftarrow \text{edge}(c_i, b_q)$, where q is randomly selected from $Q = \{q | \arg \min_q \sum_{p=0}^{i-1} g_{p,q}^2\}$. Subsequently, the largest weight coefficient w_0 is allocated to $E_{c_i}^0$ as $g_{i,q} = w_0$.

After that, the subgraph from c_i is expanded using the strategy of PEG [13], ending up with the set $\overline{N_{c_i}^l}$. When the expanded stage is finished, $E_{c_i}^j \leftarrow \text{edge}(c_i, b_q)$, where q is randomly selected from a refreshed $Q = \{q | \arg \min_q \sum_{p=0}^{i-1} g_{p,q}^2, b_q \in \overline{N_{c_i}^l}\}$. Then, the next weight coefficient w_j is allocated to $E_{c_i}^j$ as $g_{i,q} = w_j$. This step is repeated until d edges are selected, and then the process moves to the next CN.

Algorithm 1 Improved PEG-based construction algorithm.

Input: k, n, d, W

Output: AFC generator matrix \mathbf{G}

```

1:  $\mathbf{G} = \text{zeros}(n, k)$ 
2: for  $i = 0$  to  $n - 1$  do
3:   for  $j = 0$  to  $d - 1$  do
4:     if  $j = 0$  then
5:        $Q = \{q | \arg \min_q \sum_{p=0}^{i-1} g_{p,q}^2\}$ 
6:        $E_{c_i}^0 \leftarrow \text{edge}(c_i, b_q)$ , where  $q$  is randomly selected from  $Q$ .
7:       Update the  $g_{i,q} = w_0$ .
8:     else
9:       expand a subgraph from  $c_i$  up to depth  $l$  under the current graph setting such
       that the cardinality  $N_{c_i}^l$  stops increasing but  $|N_{c_i}^l| < k$ , or  $\overline{N_{c_i}^l} \neq \emptyset$  but  $\overline{N_{c_i}^{l+1}} = \emptyset$ .
10:       $Q = \{q | \arg \min_q \sum_{p=0}^{i-1} g_{p,q}^2, b_q \in \overline{N_{c_i}^l}\}$ 
11:       $E_{c_i}^j \leftarrow \text{edge}(c_i, b_q)$ , where  $q$  is randomly selected from  $Q$ .
12:      Update the  $g_{i,q} = w_j$ .
13:    end if
14:  end for
15: end for
16: Return  $\mathbf{G}$ .
```

3.3. Analysis of Improved PEG-Based Construction

Suppose that after applying Algorithm 1, a new initialized CN c_n introduces length- r cycles following the selection of $E_{c_n}^{d-e}$, where e denotes the minimum number of length- r cycles introduced by c_n . During the selection of $E_{c_n}^a$, where $d - e \leq a < d$, a selected edges $E_{c_n}^0, E_{c_n}^1, \dots, E_{c_n}^{a-1}$ during the selection of c_n , and nd edges established before the initialization of c_n are candidates that can be randomly connected to form a length- r cycle.

Denote the $r - 2$ edges of the length- r cycle established before the initialization of c_n as $(c_{j_0}, b_{i_0}), (c_{j_1}, b_{i_1}), \dots, (c_{j_{r-3}}, b_{i_{r-3}})$, where $0 \leq j < n$ and $0 \leq i < k$. Without loss of generality, denote $E_{c_n}^a \leftarrow (c_n, b_{i_0})$ as the edge closes the cycle, and denote $E_{c_n}^{a'} \leftarrow (c_n, b_{i_1})$ as the edge forming the cycle due to the presence of (c_n, b_{i_0}) , where $0 \leq a' < a$.

Moreover, divide W into two subsets according to $E_{c_n}^{d-e}$ as $W = W_{\text{large}} \cup W_{\text{small}}$, where $W_{\text{large}} = \{w_0, w_1, \dots, w_{d-e-1}\}$ and $W_{\text{small}} = \{w_{d-e}, w_{d-e+2}, \dots, w_{d-1}\}$ represent the sets consisting of relatively large and small weight coefficients, respectively. Based on the aforementioned discussion, the probabilities that a weight coefficient involved in the above edges are analyzed case by case.

Case 1:

$$P(g_{j,i} = w) = 1/d, w \in W, \tag{6}$$

where (c_j, b_i) is one of the $r - 2$ edges. The weight coefficient in (c_j, b_i) is allocated before the PEG selects edges for c_n . Thus, it is considered as randomly selected from W .

Case 2:

$$P(g_{n,i_0} = w) = 0, w \in W_{\text{large}}. \tag{7}$$

Considering that length- r cycles arise after selecting $E_{c_n}^{d-e}$, and the weight coefficients are allocated in descending order, the last selected e edges for c_n are allocated with $w \in W_{\text{small}}$. On the contrary, any $w \in W_{\text{large}}$ cannot be allocated to g_{n,i_0} .

Case 3:

$$P(g_{n,i_1} = w) = \frac{1}{a} \leq \frac{1}{d-e}, w \in W_{\text{large}}. \tag{8}$$

The corresponding (c_n, b_{i_1}) is randomly connected from the a selected edges. Thus, w can be considered as randomly selected from $\{w_0, w_1, \dots, w_{a-1}\}$.

Based on the above analysis, $r - 2$ edges (c_j, b_i) , 1 edge (c_n, b_{i_0}) , and 1 edge (c_n, b_{i_1}) have the probabilities in cases 1, 2, and 3, respectively. The probability that w is involved in an edge of the new introduced length- r cycles when $w \in W_{\text{large}}$ is

$$P(g_{j,i} = w) \leq \left(\frac{1}{d-e} + \frac{r-2}{d}\right)/r, w \in W_{\text{large}}, \tag{9}$$

where $0 \leq j \leq n$.

Example 1. Let us consider a 256×64 matrix \mathbf{G} with degree $d = 4$, which is generated using Algorithm 1. According to the parameter settings and the bound analysis in [10], a few length-4 cycles arise. Thus, we consider $r = 4$ and $e = 1$. W is divided into $W_{\text{large}} = \{w_0, w_1, w_2\}$ and $W_{\text{small}} = \{w_3\}$. Substitute d, e , and r into (9), $P(g_{j,i} = w) \leq 0.208$, when $w \in W_{\text{large}}$. Through statistical analysis of a generated \mathbf{G} , there are 25 length-4 cycles. The probabilities of weight coefficients involved in the length-4 cycles are shown in Table 1. These results verify our analysis for length-4 cycles.

Table 1. Probabilities of weight coefficients involved in the length-4 cycles.

Weight Coefficient	Probability
w_0	20%
w_1	23%
w_2	18%
w_3	39%

Example 2. Let us consider a 150×128 matrix \mathbf{G} with $d = 4$, which is generated using Algorithm 1. Similarly, we consider $e = 1$ and $r = 6$ for \mathbf{G} . W is divided in the same way as in Example 1. Substitute d, e , and r into (9), $P(g_{j,i} = w) \leq 0.22$, when $w \in W_{\text{large}}$. Through statistical analysis of a generated \mathbf{G} , there are 28 length-6 cycles. The probabilities of weight coefficients involved in the length-6 cycles are shown in Table 2. These results verify our analysis for length-6 cycles.

Table 2. Probabilities of weight coefficients involved in the length-6 cycles.

Weight Coefficient	Probability
w_0	19%
w_1	21.4%
w_2	21.4%
w_3	38.2%

3.4. Complexity Analysis

The complexity of WC-PEG is $O(nk)$ [13] in the worst case, mainly from expanding the Tanner graph. In addition to expanding the Tanner graph, our proposed algorithm involves calculating the reliability measure. It calculates d weight coefficients introduced

by the last CN with complexity $O(nd)$. Then, it sorts the d changed reliability measure with the remainder by binary insertion with complexity $O(nd \log k)$. Thus, its overall complexity is $O(n(k + d \log k))$ in the worst case. All of these algorithms take moderate complexity for constructing generator matrices, which are designed offline.

3.5. Simulation and Discussion

In this subsection, we present the numerical results of the proposed construction and the constructions in the references. Here, we select $d = 8$ and $W = \{1/2, 1/3, 1/5, 1/7, 1/11, 1/13, 1/17, 1/19\}$ for AFC constructions, which are designed in [2]. In addition, $d = 4$ and $W = \{1/2, 1/3, 1/5, 1/7\}$ are also considered. Various construction methods such as random AFC, S-AFC, and WC-PEG are included as comparison. The proposed construction is labeled as IM-PEG. These settings are in line with the previous references. The CS-BP decoder with a maximum iteration number of 15 is applied in the receiver side.

First, we simulate the BLER of different AFC constructions including random AFC, S-AFC, WC-PEG, and the proposed construction. Let $R = k/n$ denote the code rate. Simulations are conducted with fixed rate $R = 2$ and $k = 128$. Various degrees $d = 4$ and $d = 8$ are considered, respectively. The AWGN channel with zero mean and variance δ^2 is employed for the transmission of AFC codewords. Signal-noise ratios (SNRs) range from 16 dB to 24 dB. All of the parameters are set according to [2,6]. Please note that R can be larger than 1, since one AFC symbol involves d bits. It can be seen in Figure 3 that the improved PEG construction has good BLER performance. For example, when $d = 4$, the proposed construction offers about 1 dB gain when BLER is lower than 10^{-4} over WC-PEG and S-AFC.

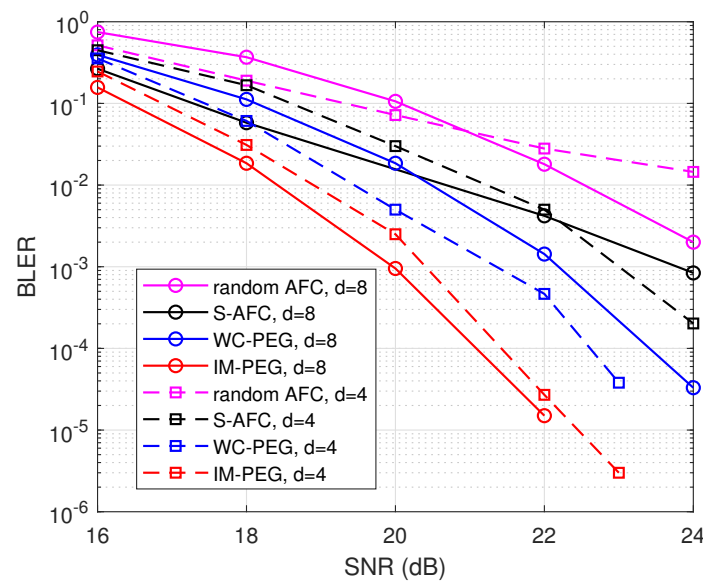


Figure 3. BLER of different AFC constructions with $d = 4$ and $d = 8$.

Then, the achievable rates of different AFC constructions with $k = 128$ are simulated. Similarly, various degrees $d = 4$ and $d = 8$ are considered, respectively. Figure 4 shows that the rates of the proposed construction are higher than others in a large range of SNRs, in both $d = 4$ and $d = 8$.

Moreover, BLER performance with various weight sets is presented in Figure 5. $k = 128, d = 4$, and $R = 1$ are set for the simulation. $W' = \{0.8670, 0.432, 0.2155, 0.1073\}$ [6] is selected for comparison. It can be seen from Figure 5 that the proposed construction provides good error performance with W' as well.

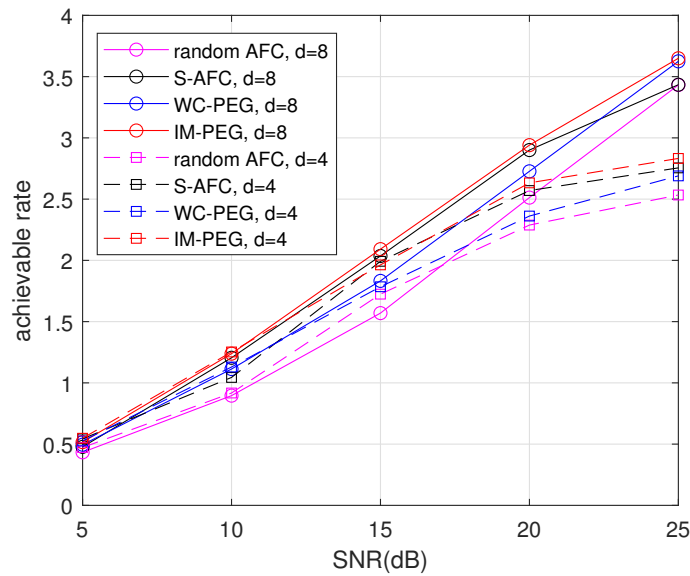


Figure 4. Achievable rates of different AFC constructions with $d = 4$ and $d = 8$.

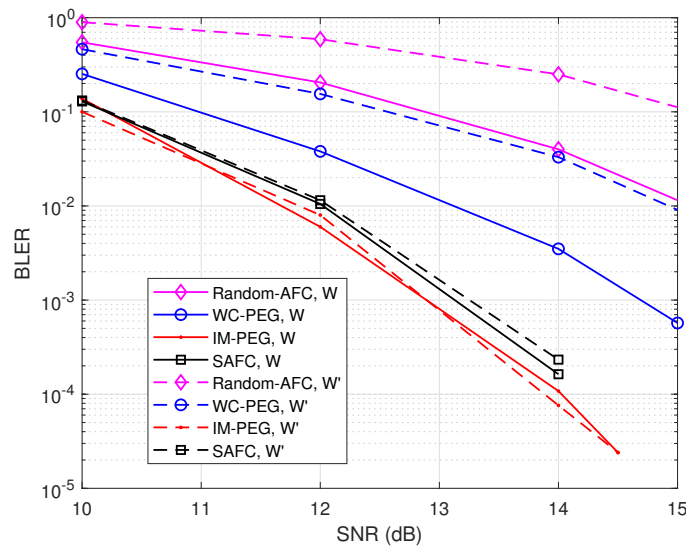


Figure 5. BLER of different weight sets with $d = 4$ and $R = 1$.

4. Conclusions

In this paper, an improved PEG-based construction of AFC is proposed to enhance decoding performance. Specifically, we propose to allocate weight coefficients in descending order simultaneously when PEG selects edges for AFC, avoiding large weight coefficients involved in the short cycles. The analysis demonstrates that the proposed construction efficiently reduces the probability that larger weight coefficients are involved in short cycles. The simulation results show that it has good BLER performance and achievable rates.

Author Contributions: Investigation, X.L. and Q.H.; Writing—original draft, X.L.; Writing—review & editing, Q.H.; Funding acquisition, Q.H. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by National Natural Science Foundation of China under Grant 62071026 and 62331002.

Institutional Review Board Statement: Not applicable.

Data Availability Statement: The original contributions presented in the study are included in the article, further inquiries can be directed to the corresponding author.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Le, T.K.; Salim, U.; Kaltenberger, F. An Overview of Physical Layer Design for Ultra-Reliable Low-Latency Communications in 3GPP Releases 15, 16, and 17. *IEEE Access* **2021**, *9*, 433–444. [[CrossRef](#)]
2. Shirvanimoghaddam, M.; Li, Y.; Vucetic, B. Near-Capacity Adaptive Analog Fountain Codes for Wireless Channels. *IEEE Commun. Lett.* **2013**, *17*, 2241–2244. [[CrossRef](#)]
3. Shirvanimoghaddam, M.; Li, Y.; Vucetic, B. Multiple access analog fountain codes. In Proceedings of the 2014 IEEE International Symposium on Information Theory, Honolulu, HI, USA, 29 June–4 July 2014; pp. 2167–2171. [[CrossRef](#)]
4. Lim, W.J.; Abbas, R.; Li, Y.; Vucetic, B.; Shirvanimoghaddam, M. Analysis and Design of Short Analog Fountain Codes for the Multiple Access Channel. *IEEE Commun. Lett.* **2022**, *26*, 1454–1458. [[CrossRef](#)]
5. Lim, W.J.; Abbas, R.; Li, Y.; Vucetic, B.; Shirvanimoghaddam, M. Analysis and Design of Analog Fountain Codes for Short Packet Communications. *IEEE Trans. Veh. Technol.* **2021**, *70*, 12662–12674. [[CrossRef](#)]
6. Abbas, R.; Shirvanimoghaddam, M.; Huang, T.; Li, Y.; Vucetic, B. Novel Design for Short Analog Fountain Codes. *IEEE Commun. Lett.* **2019**, *23*, 1306–1309. [[CrossRef](#)]
7. Zhang, K.; Jiao, J.; Huang, Z.; Wu, S.; Zhang, Q. Finite Block-Length Analog Fountain Codes for Ultra-Reliable Low Latency Communications. *IEEE Trans. Commun.* **2020**, *68*, 1391–1404. [[CrossRef](#)]
8. Li, X.; Huang, Q. Online Analog Fountain Codes. *IEEE Commun. Lett.* **2023**, *27*, 50–54. [[CrossRef](#)]
9. Lim, W.J.; Shirvanimoghaddam, M.; Abbas, R.; Li, Y.; Vucetic, B. On the Design of Analog Fountain Codes for Short Packet Communications in 5G URLLC. In Proceedings of the 2019 IEEE 90th Vehicular Technology Conference (VTC2019-Fall), Honolulu, HI, USA, 22–25 September 2019; pp. 1–5. [[CrossRef](#)]
10. Huang, Z.; Jiao, J.; Zhang, K.; Wu, S.; Zhang, Q. A Novel Weight Coefficient PEG Algorithm for Ultra-Reliable Short Length Analog Fountain Codes. *IEEE Wirel. Commun. Lett.* **2019**, *8*, 125–128. [[CrossRef](#)]
11. Zhang, K.; Jiao, J.; Wu, S.; Zhang, Q. Short Analog Fountain Code With Quasi-Gray Constellation Mapping Modulation Towards uRLLC. *IEEE Trans. Signal Process.* **2022**, *70*, 4077–4092. [[CrossRef](#)]
12. Guo, J.; Zhang, D.; Lee, I.; Li, Y.; Shirvanimoghaddam, M. Partitioned Analog Fountain Codes for Short Packet Communications. *IEEE Commun. Lett.* **2024**, *28*, 1248–1252. [[CrossRef](#)]
13. Hu, X.Y.; Eleftheriou, E.; Arnold, D. Regular and irregular progressive edge-growth tanner graphs. *IEEE Trans. Inf. Theory* **2005**, *51*, 386–398. [[CrossRef](#)]
14. Baron, D.; Sarvotham, S.; Baraniuk, R.G. Bayesian compressive sensing via belief propagation. *IEEE Trans. Signal Process.* **2009**, *58*, 269–280. [[CrossRef](#)]
15. ten Brink, S. Convergence behavior of iteratively decoded parallel concatenated codes. *IEEE Trans. Commun.* **2001**, *49*, 1727–1737. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.