A Novel Luby Transform Code with Improved Ripple Size

Dai Weina[1] and Zhao Yuli[2(✉)]

[1] College of Medicine and Biological Information Engineering, Northeastern University, Shenyang, China
20175565@stu.neu.edu.cn
[2] Software College, Northeastern University, Shenyang, China
zhaoyl@swc.neu.edu.cn

Abstract. Degree distribution of the encoded symbols is a critical factor that affects the performance of LT codes. Inspired by the SF-LT and the decreasing ripple size LT codes, we propose an SF-DRS degree distribution to construct LT codes, i.e., SF-DRS LT codes. Theoretical analysis proves that our proposed LT codes possess the property that the ripple size decreases as the decoding process continues. Moreover, with an overhead factor of slighter larger than one, the ripple size of SF-DRS LT codes remains larger than one in the entire decoding process. The performance of our proposed LT codes is compared to SF-LT codes and decreasing ripple size over a perfect channel. Simulation results reveal that the proposed SF-DRS LT codes outperform the decreasing ripple size LT codes with respect to the probability of successful decoding, the average overhead factor with a relatively large number of input symbols. Moreover, in contrast to the SF-LT codes, the SF-DRS LT codes achieve a better probability of successful decoding.

Keywords: Luby transform codes · Fountain codes · Degree distribution

1 Introduction

It has been proven that fountain codes, including Luby transform codes [1, 2], raptor codes [3], online codes [4, 5], could achieve excellent performance over time-varying channels and highly impaired channels [6]. Fountain codes could potentially be applied in the field of scalable video transmission [7], deep communications [8], and wireless relay networks [9, 10].

Luby transform codes, LT code for short, is a typical type of fountain codes. Given K input symbols, LT codes could generate an unlimited number of encoded symbols. When slighted larger than K encoded symbols are received, the decoder could successfully recover the original input symbols [6]. The degree distribution and the short-cycle structure in the Tanner graph are two factors that affect the performance of LT codes the most. The LT codes using ideal soliton degree distribution theoretically emerge one degree-1 encoded symbol at each iteration in the decoding process [1]. However, due to

H. Song and D. Jiang (Eds.): SIMUtools 2020, LNICST 370, pp. 649–658, 2021.
https://doi.org/10.1007/978-3-030-72795-6_52

the randomness, any small fluctuation of the number of degree-1 encoded symbols will lead to the failure of the whole decoding process. Therefore, LT codes based on ideal soliton distribution are not available in practice [3]. The LT codes using robust solution degree distribution are the de facto codes discussed in recent researches.

Based on the robust soliton distribution, by adjusting the proportion of degree-1, degree-2 and the maximum degree, Yen et al. derive a modified robust soliton distribution [11]. The LT codes using modified robust soliton distribution is aimed at increasing the mean of the expected ripple size to improve the ripple evolution. In [12], Zhu et al. theoretically analyze the LT BP (belief propagation) decoding process and propose an optimal degree distribution algorithm. Based on this for simplicity, a "suboptimal" degree distribution is proposed to improve the coding and decoding performance. Zhao et al. analyze the LT decoding process with the perspective of complex networks and propose a scale-free Luby transform code, SF-LT code for short [2]. This code is based on the fact that the iteration process between input symbols and encoded symbols in the Tanner graph can be seen as a message traveling process in a complex network. Complex networks with scale-free degree distribution have the shortest path length compared with other similar scale-size complex networks. In [13], Savchenko et al. propose using deep reinforcement to learn an optimal degree distribution of LT based code. In [14, 15], Jesper et al. have proved that the ripple size of a well-performed LT code should decrease with the decoding iteration process. The degree distribution proposed by the authors could further reduce the number of encoded symbols required for decoding and significantly improving the performance.

Other schemes combing improved degree distribution and the selection of input symbols when generating an encoded symbol are also proposed. In [16], Yen et al. show that in LT code, most of the decoding termination is caused by no ripple in the early stage. Hence, the authors modify the robust Soliton distribution and introduce a non-repetitive encoding scheme to avoid repeated degree-1 encoded symbols. This scheme requires a smaller number of encoded symbols to get a larger decoding probability. In [17], the authors modify the encoding process to maximize the minimum degree of the input symbols. This scheme improves the robustness of the LT codes over erasure channels. However, the encoding complexity increases with the enlarged average degree of the encoded symbols.

Inspired by scale-free LT codes proposed in [2] and LT codes with decreasing ripple size, we combine the modified scale-free distribution and the degree distribution proposed in [14] to construct a novel LT code with balanced decreasing ripple size. The proposed method could improve the decoding performance including average overhead factor and probability of successful decoding. Moreover, the complexity also outperforms the codes in [14]. The rest of this paper is organized as follows. Section 2 reviews the principle of LT codes. Section 3 provides the degree distribution of the encoded symbols we proposed and its theoretical analysis. Simulation results are shown in Sect. 4. Finally, Sect. 5 concludes the paper.

2 Review of the LT Codes

Given the set of input symbols $\{s_1, s_2, \ldots, s_{K-1}, s_K\}$ and the degree distribution of the encoded symbols $\Omega(\mathrm{x})$, the ith encoded symbol c_i is generated as follows.

1) Select a degree according to the degree distribution and denote it as d_i.
2) uniformly select d_i input symbols $\{s_{i1}, s_{i2}, s_{i3}, \ldots s_{idi}\}$;
3) the encoded symbol c_i is assigned with $\sum_{j=1}^{d_i} s_{ij}$.

Repeat step 1)–3), encoded symbols can be continually generated. Then, taking the input symbols as one node set and the encoded symbols as the other node set, a tanner graph can be constructed according to the encoding process. Only if input symbol s_i is taken as one selected symbol for generating encoded symbol c_j, there exists an edge between node s_i and c_j in the Tanner graph. Figure 1 shows a Tanner graph with 5 input symbols and 6 encoded symbols.

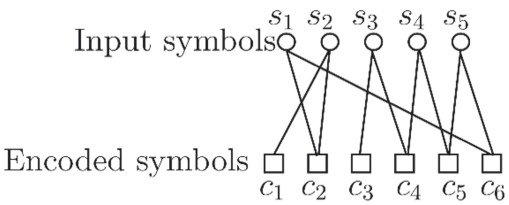

Fig. 1. A Tanner graph with 5 input symbols and 6 encoded symbols.

The BP decoding process over the erasure channel could be considered as a message-passing process. When N encoded symbols have been received (N is slightly larger than K), the decoder starts to recover the input symbols. A particular set that contains all degree-1 encoded symbols is defined as a ripple. The BP decoding process could be described as follows.

1) find all degree-1 encoded symbols and put them in the ripple set;
2) randomly select an encoded symbol c_j from the ripple set.
3) the neighbor input symbol (s_i) of the selected encoded symbol is recovered immediately by copying the value of c_j to the input symbol. The edge that connects these two symbols is removed from the tanner graph.
4) updating the value of the encoded symbols that connect with the newly recovered input symbol by obtaining the XOR value of the encoded symbol and the input symbol; remove the edges connecting the newly recovered input symbol and its neighboring encoded symbols.
5) repeat 1)–4), until all input symbols have been recovered. Then, the decoder informs the successful decoding event to the sender.

3 Proposed SF-DRS LT Code and Its Theoretical Analysis

3.1 SF-DRS Degree Distribution

In [14], Jesper et al. proved that an encoded symbol with a higher degree is more likely to be redundant than lower degree ones. Early in the decoding process, the low degree encoded symbols decrease their degree to one. At this stage, an input symbol has a

relatively low probability of being repeatedly added to the ripple set. However, when high degree encoded symbols are becoming degree-1 ones late in the decoding process, the input symbol neighbored with the degree-1 encoded symbol is more likely existing in the ripple set already. Thus, the ripple size should be decreasing as the decoding process continues. Moreover, it has been proven that in contrast to other same scale networks, scale-free networks possess the lowest short-length path property. But, the ripple size of LT codes using modified scale-free degree distribution does not decrease with the iteration of the decoding process.

In this section, we add the degree distribution proposed in [14] to the scale-free degree distribution in [2] and obtain an SF-DRS distribution. The objection of the addiction is to get a distribution both possessing scale-free property and having a decreasing ripple size when decoding process continues. The scale-free distribution [2] is given by

$$\lambda(d) = \begin{cases} P_1, & d = 1, \\ A \cdot d^{-\gamma}, & d = 2, 3, \cdots K-1, K. \end{cases} \tag{1}$$

Where A is the normalization coefficient, γ is the characteristic index, and P_1 is the fraction of degree-1 encoded symbols. Moreover, the parameters in (1) satisfy $\sum_{d=1}^{K} \lambda(d) = 1$.

The degree distribution of LT codes proposed by Jesper in [14] is given by

$$\begin{cases} \theta(1) = \frac{R}{n}, \\ \theta(2) = \frac{K(K-1)}{2n(K-R)}, \\ \theta(i) = \frac{i-1}{i}\theta(i-1), \quad i < \frac{K}{3}, \\ \theta(i) = \theta(i-1), \quad \frac{K}{3} \le i < \frac{2K}{3}, \\ \theta(i) = \frac{K-i+1}{K-i}\theta(i-1), \quad \frac{2K}{3} \le i \le K-R+1. \end{cases} \tag{2}$$

Where the parameters n and R should satisfy $\sum_{d=1}^{K} \theta(d) = 1$.

Our proposed SF-DRS distribution is obtained by normalizing (1) and (2), and expressed as

$$\varphi(d) = \frac{\lambda(d) + \theta(d)}{\sum_{t=1}^{K} \lambda(t) + \theta(t)}, \quad d = 1, 2, 3, \ldots, K. \tag{3}$$

The LT codes which degree distribution of encoded symbols follows the SF-DFS distribution is named as SF-DFS LT codes. Figure 2 shows the degree distribution curve of encoded symbols in an SF-DFS LT codes when $P_1 = 0.1$, $\gamma = 1.9$, $n = 1049$, $R = 20$, $K = 1000$. It can be seen that a large number of encoded symbols are low degree nodes. Hence, the decoder could start decoding and iteratively recover the input symbols. With some high degree encoded symbols, we can guarantee that all the input symbols participate in the generation process of encoded symbols.

3.2 Ripple Set Analysis

Define overhead factor α as the ratio of the number of encoded symbols and the number of input symbols. Assume N encoded symbols are received, the overhead factor is calculated

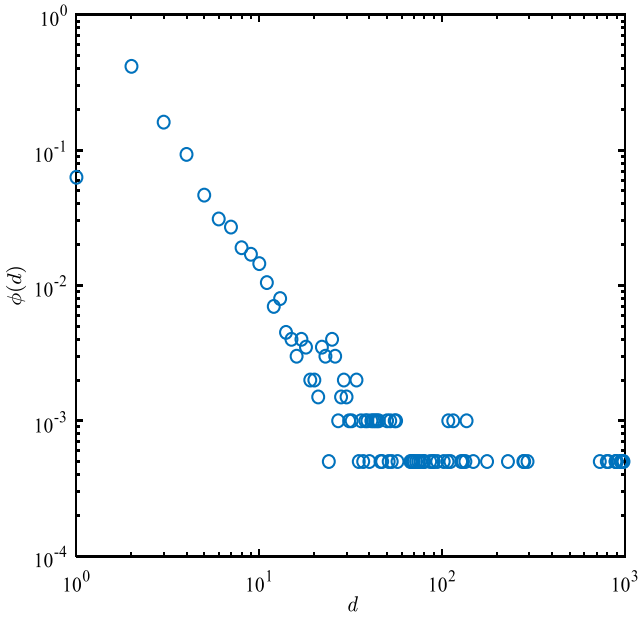

Fig. 2. Degree distribution of encoded symbols in SF-DFS LT codes. $P_1 = 0.1$, $\gamma = 1.9$, $n = 1048$, $R = 20$, $K = 1000$.

by $\alpha = N/K$. Given the degree distribution of the encoded symbols, the evolution of the ripple size [14] can be calculated by

$$\begin{cases} \eta^K(i) = \alpha K \varphi(i), i = 1, 2, \ldots, K; \\ \eta^{L-1}(1) = \eta^L(1) - 1 + \frac{2\left(L - \eta^{(L)}(1)\right)}{L(L-1)} \eta^{(L)}(2) \\ \eta^{L-1}(i) = \eta^L(i) - \frac{i}{L}\eta^L(i) + \frac{i+1}{L}\eta^L(i+1), i = 2, 3, \ldots, L-1; \\ \eta^{L-1}(L) = 0. \end{cases} \tag{4}$$

Where $\phi(i)$ is the degree distribution of encoded symbols; $\eta^L(i)$ is the number of degree-i encoded symbols left when L input symbols are not recovered. Equation (4) is based on the assumption that each input symbol only has one opportunity to be added into the ripple set. However, due to the randomness of the connection between encoded symbols and the input symbols, it could not guarantee that the input symbols were put into the ripple set without repetition. Theoretically, the codes with ripple size larger than 1 from $L = 1$ to $L = K$ could recover the input symbols successfully. Hence, setting $\alpha = 1.1$, we substitute (3) to (4), and get the theoretical ripple evolution process. Figure 3 illustrates the theoretical ripple evolution of SF-DRS LT codes and LT codes based on robust soliton distribution. It can be seen that both SF-DRS LT codes ($P_1 = 0.1$, $\gamma = 2.0$) and SF-DRS LT codes ($P_1 = 0.1$, $\gamma = 1.9$) achieve ripple size of larger than 1. Moreover, compared with LT codes using robust soliton degree distribution, these two codes achieve a larger ripple size at the beginning phase of the decoding process. The ripple size of the SF-DRS LT codes ($P_1 = 0.1$, $\gamma = 2.0$) decreases as the decoding process continues. Thus,

theoretically, when $N = \alpha K = 1.1K$ encoded symbols received, the SF-DRS LT codes could recover all input symbols successfully.

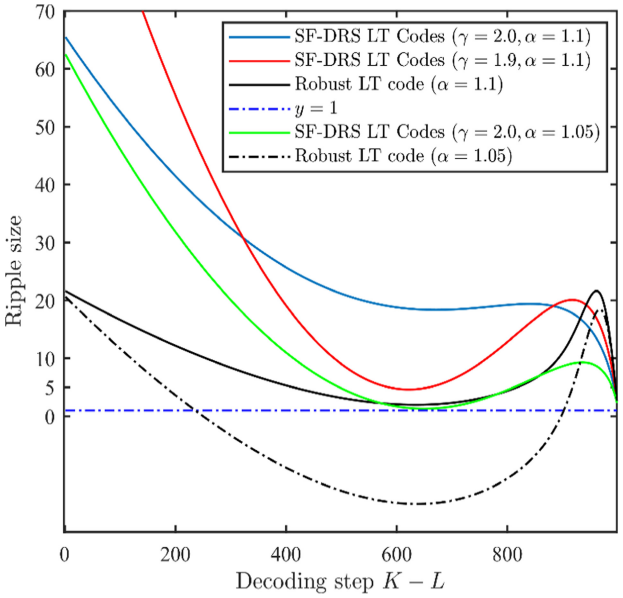

Fig. 3. The ripple evolution of the proposed LT codes. $K = 1000$.

Moreover, when the overhead factor decreases to 1.05, the ripple size of robust LT codes becomes smaller than one at some points. It indicates that using robust LT codes, the decoder could not recover all input symbols when 1050 encoded symbols are received. Inversely, the SF-DRS LT codes ($P_1 = 0.1$, $\gamma = 2.0$) always achieve a ripple size of larger than one at each iteration. Hence, when the receiver gets 1050 encoded symbols, the SF-DRS LT codes ($P_1 = 0.1$, $\gamma = 2.0$) could decode all the 1000 input symbols successfully.

4 Simulation Result

In this section, we assume each input symbol contains $l = 1$ bit, and randomly generate $K = 1000$ ($K = 2000$) input symbols. Then, we use the following codes to encode each group of input symbols.

(i) SF-DRS1 LT code: our proposed SF-DRS LT code with $P_1 = 0.1$, $\gamma = 2.0$, $n = 1049$, $R = 20$;
(ii) SF-DRS2 LT code: our proposed SF-DRS LT code with $P_1 = 0.1$, $\gamma = 1.9$, $n = 1049$, $R = 20$;
(iii) Decreasing ripple size LT code: LT code proposed in [14] with $n = 1049$, $R = 20$ (when $K = 1000$); when $K = 2000$, parameters are $n = 2056$, $R = 24$;

(iv) SF-LT1 code: the LT code proposed in [2] using parameters $P_1 = 0.1$, $\gamma = 2.0$;
(v) SF-LT2 code: the LT code proposed in [2] using parameters $P_1 = 0.1$, $\gamma = 1.9$.

As we consider the effect of the number of received encoded symbols, we ignore the erasure probability of the erasure channel, and assume the generated encoded symbols are transmitted over a perfect channel, i.e. the erasure probability is zero. For each LT code, as the receiver continuously obtain encoded symbols and recover parts of the input symbols, we record the number of input symbols being recovered. Figure 4 shows the probability of successful decoding versus overhead factor when $K = 1000$.

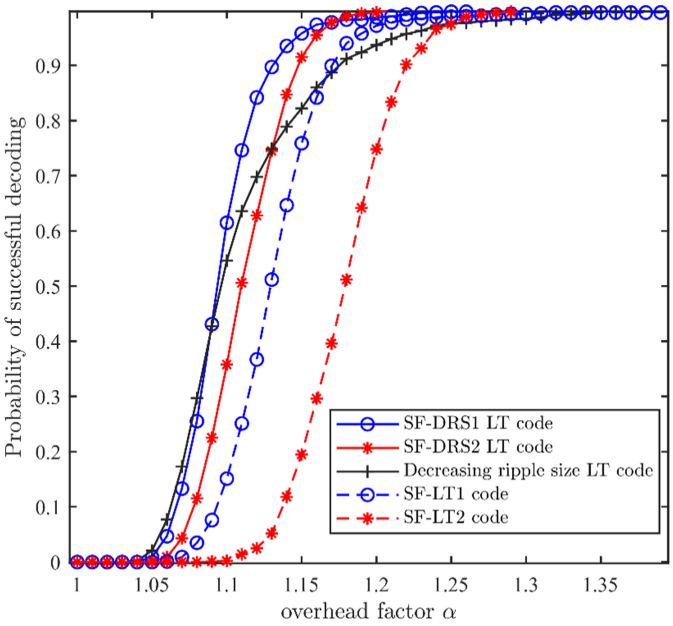

Fig. 4. Probability of successful decoding versus overhead factor ($K = 1000$)

It can be seen from Fig. 4 that when enough encoded symbols have been received, the receiver begins to recover the original input symbols. Initially, the decreasing ripple size LT codes start to decode some input symbols prior to other LT codes. As encoded symbols continuously being delivered to the receiver, SF-DRS1 LT codes outperform decreasing ripple size LT codes with respect to the immediate probability of successful decoding when the overhead factor is larger than 1.09. Moreover, in contrast to the SF-LT codes, the SF-DRS LT codes with the same parameters achieve a larger probability of successful decoding.

Figure 5 shows the probability of successful decoding versus overhead factor when $K = 2000$. When the overhead factor increases to 1.08 (1.12), the performance of SF-DRS1 (SF-DRS2) LT codes surpasses that of the decreasing ripple size LT codes. Moreover, when the overhead factor approaches 1.16, the SF-DRS LT codes and SF-LT1 codes recover 99% input symbols. However, even when the overhead factor increases to 1.24,

the decreasing ripple size LT codes could recover 98.5% input symbols. With the same parameters, SF-DRS LT codes achieve a much better probability of successful decoding than SF-LT codes.

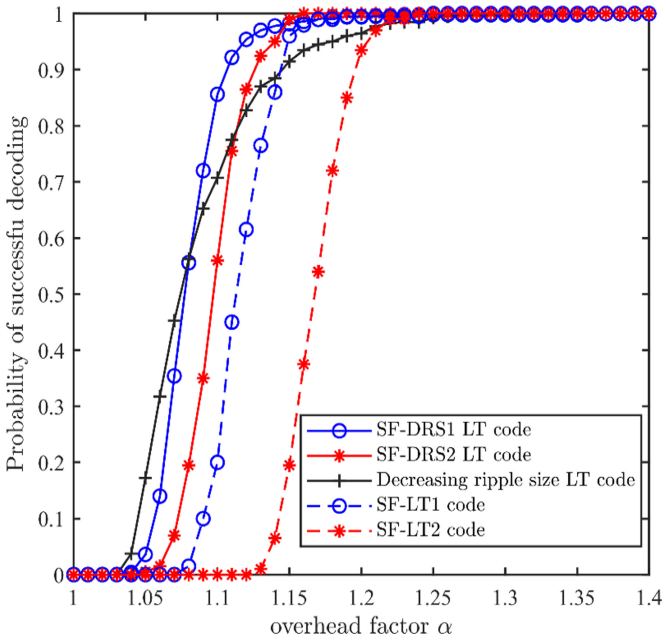

Fig. 5. Probability of successful decoding versus overhead factor ($K = 4000$)

For each type of LT codes, we independently run 1000 times for $K = 500, 1000, 2000$, and record the overhead factor for recovery all input symbols. Denote α_i as the overhead factor of i-th independent simulation, the average overhead factor can be calculated by

$$\overline{\alpha} = \sum_{t=1}^{1000} \alpha_i \tag{5}$$

Figure 6 illustrates the average overhead factor of SF-DRS1 LT codes, SF-DRS2 LT codes and decreasing ripple size LT codes when $K = 500, 1000, 2000$. It can be seen that for the three LT codes, the average overhead factor decreases with the increase of K. Moreover, SF-DRS1 LT code achieves the minimum average overhead factor compared with SF-DRS2 LT code and decreasing ripple size LT code. When $K = 500$, the decreasing ripple size LT code requires less number of encoded symbols to recover the input symbols than the SF-DRS2 LT code. However, when K increases to 1000 and 2000, the SF-DRS2 LT code outperforms the decreasing ripple size LT code in terms of the average overhead factor.

For further evaluating the efficiency of our proposed LT codes, we calculate the average degree of the encoded symbols using (6).

$$\overline{d} = \sum_{d=1}^{K} d \cdot \varphi(d) \tag{6}$$

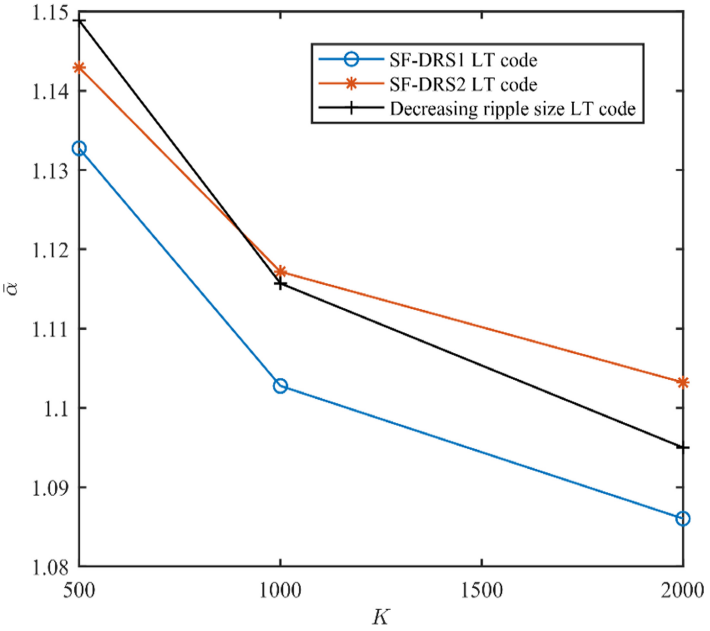

Fig. 6. Average overhead factor of SF-DRS1 LT code, SF-DRS2 LT code and Decreasing ripple size LT code. $K = 500, 1000, 2000$.

The average degrees of SF-DRS1 LT codes are 13.84 ($K = 1000$) and 15.41 ($K = 2000$), which are smaller than the average degrees of the decreasing ripple LT codes (15.32, when $K = 1000$; 17.56, when $K = 2000$). Thus, the coding efficiency of SF-DRS1 LT codes is superior to that of the decreasing ripple size LT codes.

5 Conclusion

In this paper, we combine the advantage of SF-LT codes and decreasing ripple size LT codes, and propose an LT code using SF-DRS degree distribution. Using the ripple evolution formula, we theoretically analyze the overhead factor of the SF-DRS LT codes. By selecting appropriate parameters, the ripple size of the SF-DRS LT codes continuously decreases with the decoding process. Simulation results further show that the SF-DRS1 LT codes obtain better performance than SF-DRS2 codes, SF-LT codes, and decreasing ripple size LT codes.

Acknowledgment. This research was supported by the National Natural Science Foundation of China (Grant Nos. 61603082, 61977014, 61902056 and 61902057), and the Fundamental Research Funds for the Central Universities (Grant Nos. N2017016).

References

1. Luby, M.: LT codes. In: Proceeding of the 43rd Annual IEEE Symposium on Foundations of Computer Science, Vancouver, BC, Canada, pp. 271–280. IEEE (2002)
2. Zhao, Y.L., Lau, F.C.M., Zhu, Z.L., Yu, H.: Scale-free Luby transform codes. Int. J. Bifurcat. Chaos **22**(4), 1250094-1–1250094-11 (2012)
3. Shokrollahi, A.: Raptor codes. IEEE Trans. Inf. Theory **52**(6), 2551–2567 (2006)
4. Cassuto, Y., Shokrollahi, A.: Online fountain codes with low overhead. IEEE Trans. Inf. Theory **61**(6), 3137–3149 (2015)
5. Zhao, Y.L., Zhang, Y., Lau, F.C.M., Yu, H., Zhu, Z.L.: Improved online fountain codes. IET Commun. **12**(18), 2297–2304 (2018)
6. MacKay, D.J.C.: Fountain codes. IEE Proc. Commun. **152**(6), 1062–1068 (2005)
7. Yuan, L., Li, H.A., Yi, W.: A novel UEP fountain coding scheme for scalable multimedia transmission. IEEE Trans. Multimed. **18**(7), 1389–1400 (2016)
8. Fang, J.C., Bu, X.Y., Yang, K.: Retransmission spurts of deferred NAK ARQ in fountain coding aided CCSDS file-delivery protocol. IEEE Commun. Lett. **20**(4), 816–819 (2016)
9. Nessa, A., Kadoch, M.: Joint network channel fountain schemes for machine-type communications over LTE-advanced. IEEE Internet Things J. **3**(3), 418–427 (2016)
10. Baik, J., Suh, Y., Rahnavard, N., Heo, J.: Generalized unequal error protection rateless codes for distributed wireless relay networks. IEEE Trans. Commun. **63**(12), 4639–4650 (2015)
11. Yen, K.K., Liao, Y.C., Chen, C.L., Chang, H.C.: Modified robust soliton distribution (MRSD) with improved ripple size for LT codes. IEEE Commun. Lett. **17**(5), 976–979 (2013)
12. Zhu, H.P., Zhang, G.X., Li, G.X.: A novel degree distribution algorithm of LT code. In: Proceeding of the 11th IEEE International Conference on Communication Technology, Hangzhou, China, pp. 221–224. IEEE (2008)
13. Savchenko, Y., Liu, Y.: Optimizing degree distributions of LT-based codes with deep reinforcement learning. In: IEEE Conference on Computer Communications Workshops, IEEE INFOCOM 2019, Paris, France, pp. 228–233. IEEE (2019)
14. Sørensen, J.H., Popovski, P., Ostergaard, J.: On LT codes with decreasing ripple size. In: Information Theory and Applications Workshop (ITA), La Jolla, USA (2011)
15. Sørensen, J.H., Popovski, P., Ostergaard, J.: Design and analysis of LT codes with decreasing ripple size. IEEE Trans. Commun. **60**(11), 3191–3197 (2012)
16. Yen, K.K., Liao, Y.C., Chen, C.L., Zao, J.K., Chang, H.C.: Integrating non-repetitive LT encoders with modified distribution to achieve unequal erasure protection. IEEE Trans. Multimed. **15**(8), 655–658 (2013)
17. Hussain, I., Xiao, M., Rasmussen, L.K.: Design of LT codes with equal and unequal erasure protection over binary erasure channels. IEEE Commun. Lett. **17**(2), 261–264 (2013)