Journal of Electrical Engineering, Vol. 75, No. 3, 2024, pp. 192-197

sciendo

PAPERS

Scalable codes with locality and availability derived from tessellation via [7, 3, 4] Simplex code graph

Peter Farkaš[1,2]

A new family of scalable codes with locality and availability for information repair in data storage systems for e-health applications was presented recently. The construction was based on a graph of the [7, 3, 4] Simplex code. In this paper it is shown that the construction can be generalized via tessellation in a Euclidian plane. The codes obtained have new interesting recoverability properties. They can in some cases repair damage to many storage nodes in multiple connected graphs via sequential decoding, which is similar to healing wounds in biological systems. The advantages of the original codes, namely the availability, functionality, efficiency and high data accessibility, will be preserved also in these new codes. The computational complexity and communication costs of their incrementation will remain constant and modest. These codes could be adapted to disaster recovery because it is straightforward to place the nodes so that the graph is easily mapped on a real structure in space.

Keywords: codes with locality and availability, error correction codes, scalability, distributed storage, redundancy

1 Introduction

The storing, processing and analytics of the data collected from connected devices is changing healthcare dramatically. In information systems for e-health applications information has to be stored not only securely but also reliably, with high availability, functionality, efficiency and high data accessibility [1]. However, the main challenge in this area is the volume of data that must be protected. In [2] it was stated that healthcare companies experienced, just 3 years before this publication, a 900% increase in data volume, which they have to deal with. Therefore, the storage systems for healthcare data should be scalable and the cost of this scalability has to be restricted in order to make them economically sustainable. To keep the costs low, extending the storage volume should be computationally simple and the volume of data which has to be communicated during this has to be restricted.

The storage system in its present form is usually composed of a large number of servers distributed in space. As a rule, some servers break from time to time and the information which they store has to be protected against loss using some redundant servers [3]. The simplest approach is to make more copies of the data. The drawback of this technique is that it requires relatively high redundancy and costs. A better technique is to use erasure correcting codes. This can significantly decrease the required redundancy or even minimize it in cases where so called Maximum Distance Separable (MDS) codes are used. Therefore, Reed Solomon codes with erasure decoding are very popular nowadays in practical storage such as the Hadoop Distributed File System, Google's Colossus, Yahoo Object Store, Quantcast File System, Facebook's f4, Baidu's Atlas and Backblaze's Vaults [4]. The introduction of storage distributed in space invoked new challenges and interest in constructing new codes with different properties for data recovery. If the datacenter has a distributed character, then the resources (or costs) which have to be spent for data recovery contain not only redundant storage nodes but also the communication bandwidth needed during data repair [5-7]. The actual research how to achieve it is directed in three basic directions. The first is oriented on decoding methods, which minimize the repair bandwidth for practical Reed Solomon Codes [8-12]. The second is focused on constructing new codes, which are adapted from scratch to this requirement, so called Locally Repairable Codes (LRC) [13-17]. The third is a hybrid approach between the previous two [18]. Recently it was discovered that beyond locality it is also desirable to implement availability into codes for data recovery in distributed storage systems [20-25]. One approach how to construct such codes is based on graphs or hypergraphs [26-28]. Using a graph of the [7, 3, 4] Simplex code in [29] a new family of codes with locality and availability was proposed, namely with scalability.

[1] Institute of Multimedia Information and Communication Technologies, Faculty of Electrical Engineering and Information Technology, Slovak University of Technology, Ilkovičova 3, 812 19 Bratislava, Slovakia
[2] Institute of Applied Informatics, Faculty of Informatics, Pan-European University, Tematínska 10, 851 05 Bratislava, Slovakia
p.farkas@ieee.org

https://doi.org/10.2478/jee-2024-0023, Print (till 2015) ISSN 1335-3632, On-line ISSN 1339-309X
© This is an open access article licensed under the Creative Commons Attribution-NonCommercial-NoDerivatives License (http://creativecommons.org/licenses/by-nc-nd/4.0/).

In this paper a generalization of these codes is presented using two-dimensional (2D) Euclidian plane tessellation. The obtained codes have interesting regeneration properties. Namely they are able to regenerate in some cases a high number of lost nodes (erased symbols), as long as they are restricted in some 2D area and some other conditions are fulfilled. In this paper the new property is denoted as "wound healing" because it has some similarity to this process in bio-logical systems.

The paper is organized as follows. Section 2 contains a very brief basic introduction to the theory of linear block codes, codes with locality, codes with availability and codes with locality and availability is given. Section 3 introduces the new family of scalable code with locality and availability obtained by tessellation of the two-dimensional Euclidian plane. Section 4 contains concluding remarks and a note on further research which can be done in this area in future.

2 Linear block codes with locality, availability and scalability

2.1 Linear block codes

A linear block code C is a k-dimensional subspace of an n-dimensional vector space over a finite field $GF(q)$. Usually, the codes are denoted using their basic parameters as $[n, k, d_m]$ codes, where n is the codeword length and k is the number of information symbols which this codeword transports. The third parameter d_m is a code distance, which is the minimal Hamming distance $d_m = min\{d(\mathbf{c}_i, \mathbf{c}_j\}$ between any two codewords $\mathbf{c}_i \in C$ and $\mathbf{c}_j \in C$. The Hamming distance of two codewords is the number of symbols by which these two codewords differ.

2.2 Codes with locality

Codes with locality allow correction of erased symbols using only a small number of other (local) symbols. More precisely, the i-th symbol c_i; $1 \leq i \leq n$ of a codeword $\mathbf{c} \in C$, where C is an $[n, k, d_m]$ linear block code, has locality r if c_i can be recovered by accessing at most r other symbols from $\mathbf{c}$.

An $[n, k, d_m]$ code C has locality r and is denoted as r-LRC if and only if all codeword symbols of all its codewords have locality r.

2.3 Codes with availability

Availability is a property which allows recovering one erased codeword symbol in more than one way by accessing different disjoined sets of codeword symbols. More precisely, the i-th symbol c_i; $1 \leq i \leq n$ of a codeword $\mathbf{c} \in C$, where C is an $[n, k, d_m]$ linear block code, has availability t if and only if each c_i can be recovered in t ways by accessing at least t disjoined sets (called repair sets) of other symbols in that $\mathbf{c}$. An $[n, k, d_m]$ linear block code has availability t if and only if each c_i can be recovered in t ways by accessing at least t dis-joined sets of other symbols in each $\mathbf{c} \in C$. It is denoted as t-LRC if and only if all codeword symbols of all its codewords have availability t.

2.3.1 Codes with locality and availability

The i-th symbol c_i; $1 \leq i \leq n$ of a codeword $\mathbf{c} \in C$, where C is an $[n, k, d_m]$ linear block code, has locality r and availability t if and only if each c_i can be recovered in t ways by accessing at least t disjoined repair sets of other symbols in that $\mathbf{c}$ and each repair set has at most r symbols. An $[n, k, d_m]$ linear block code has locality r and availability t if and only if each c_i can be recovered in t ways by accessing at least t disjoined sets of other symbols in each $\mathbf{c} \in C$. Such a code is usually denoted as (r, t)-LRC in literature.

2.4 Codes with locality, availability and scalability

The scalable codes with locality and availability denoted as S-(r, t) are (r, t)-LRC, which allows increasing the protected memory volume in such a way that the basic parameters r and t will not be degraded.

From a practical point of view it is also desirable that the scalability of S-(r, t) codes is not very complex computationally and the demands on communication are not very high. It is also practical if they remain constant for each increased unit of additional memory volume.

In Fig. 1, a family of S-$(2, t > 3)$ codes are illustrated using graphical representation, which was obtained in [29] from a [7, 3, 4] Simplex code which is in fact a (2, 3)-LRC.

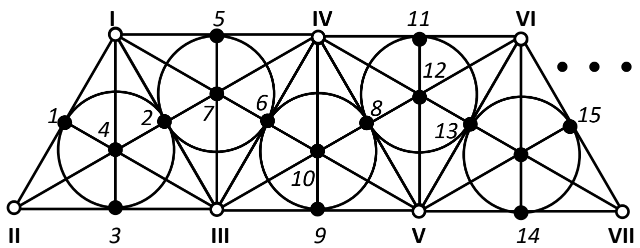

Fig. 1. Graphically represented family of S-$(2, t > 3)$ codes. The following simplified notation is used: Roman numerals denote information symbols I, II, III, IV, V, VI, VII and Arabic numbers denote parity symbols: *1*, *2*, *3*, *4*, *5*, *6*, *7*, *8*, *9*, *10*, *11*, *12*, *13*, *14*.

The storage volume increase (incrementation) which was described in [1] in more detail could be repeated indefinitely. It is obvious that the prevalent part of information symbols will have availability t=7. Only the leftmost and rightmost information symbols will have t=3 (in Fig. 1, these are symbols denoted as II and VII) and the information nodes which are second from each end will have t=5 (in Fig. 1, these are symbols denoted as I and VI). The incrementation will not decrease the locality and availability of the original [7, 3, 4] Simplex code.

3 Codes with locality, availability and scalability obtained by 2D Euklid's plane tessellation

The simplest approach to generalize the construction of S-(2, $t \geq 3$) codes described in the previous section is illustrated in Fig. 2.

The idea is to tessellate the two-dimensional (2D) Euclidian space (plane) with a pattern composed of connected graphs of the original [7, 3, 4] Simplex code as illustrated in Fig. 2. In it the black circles correspond to servers containing redundancy and the white circles to the servers containing information. In the depicted fragment of the plane there are 81 nodes containing information (payload) and 320 nodes containing redundancy. Therefore, n=401, k=81 and the overall coderate $R_c \approx 0.2$. All nodes have the locality equal to two and minimal availability three.

It is obvious that in general, during storage volume increase in the new codes family the availability and locality will not be degraded. The computational and communication expenses will also remain modest and restricted similarly to the family of codes described in [29].

Most of the information nodes which are not on border of the fragment have availability equal to twelve.

This construction also has some other advantages in practice: It has a clear structure which makes it easy to keep an overview during data center set up (placement of servers), its maintenance and makes it easy to increase the storage volume (theoretically indefinitely). The structure allows making the code easily location-conscious and consequently suitable for disaster recovery. Related information on coding for storage for disaster recovery could be found in [30]. This advantage can be synergistically strengthened by the following observations.

A not so obvious advantage of the new family of codes is their ability to recover a significant number of nodes in cases they are lost in restricted areas of the tessellation pattern. For example, if the lost nodes are inside the area bordered by dashed lines, as illustrated in Fig. 3 they all can be recovered.

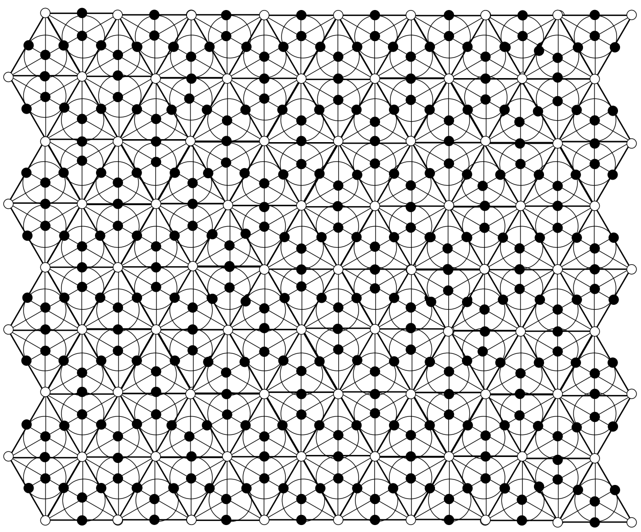

Fig. 2. Illustration of the new family of 2D- S-(r, t) codes

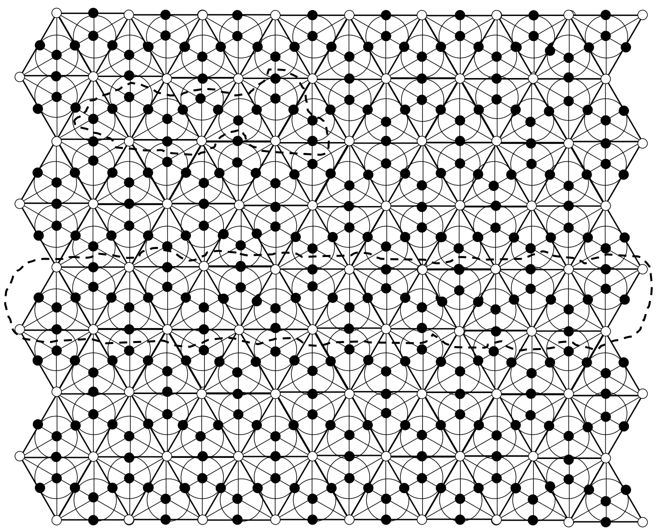

Fig. 3. Illustration of two wounds – restricted areas in which all nodes are lost or damaged (forming connected graphs). The areas are inside the dashed lines here.

For example, the regeneration of the lost nodes inside the restricted upper area (wound) in Fig. 3 can be realized in alphabetical order as illustrated in Fig. 4. The wound healing can be done using sequential decoding [31, 32] or sequential and parallel decoding [33].

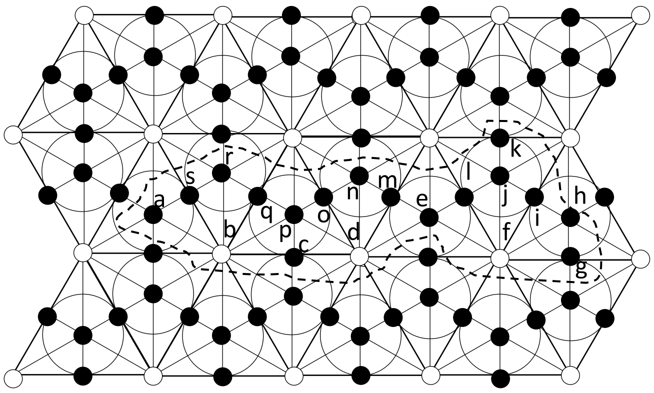

Fig. 4. Example of sequential decoding (wound healing) in the proposed 2D- S-(r, t) code.

The question arises how to proceed in regeneration of a wound in the proposed codes in general? One straightforward heuristic approach, which could be used, is to start the regeneration on the border between lost and non-lost nodes and restore each node, which is at least in one check equations with two non-lost nodes. The regeneration attempts could go as deep into the center of the wound as possible. When this direction to the center will not allow us to restore other nodes, then movement along the border, for example counterclockwise could be attempted. After finishing one cycle we can continue with the next one. The algorithm can stop if one cycle is made without regenerating any single node.

After finishing there could be still some nodes not regenerated and in this case we can say that decoding was not successful, but in practice, even in such a case, lots of nodes could be restored.

Observing the specific example of wound healing property in Fig. 4, a natural question can be asked: "How many nodes can be regenerated in one wound"? The question is not possible to answer easily in general case, because it depends on its form and even on its position with respect to other wounds as will be illustrated later. Even to answer a simpler question when a node cannot be restored is quite involved but it can be answered for some cases. The examples in Fig. 5 and Fig. 6 illustrate why we have to be careful when looking for the answer on the above question. In Fig. 5 we can see that in the wound circumscribed with punctured line we cannot regenerate even one single node. In contrast, in the wound bordered by punctured line in Fig. 6 it is possible to regenerate all nodes. It is remarkable that in Fig. 6 the wound contains only two less nodes than the wound in Fig. 5.

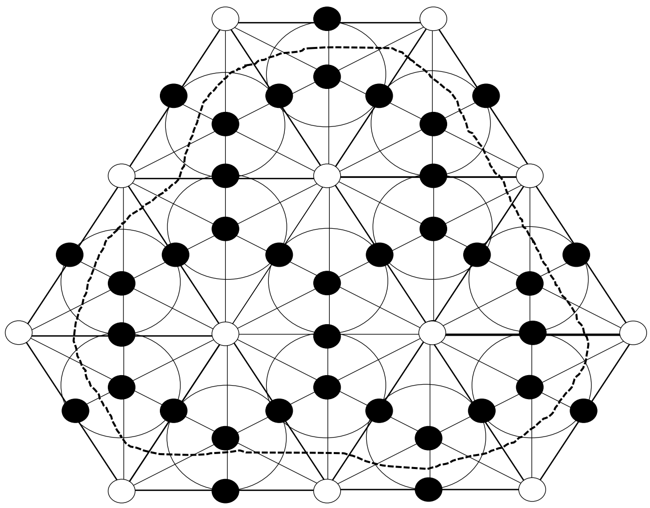

Fig. 5. Example of an unhealable wound in the proposed 2D- S-(r, t) code

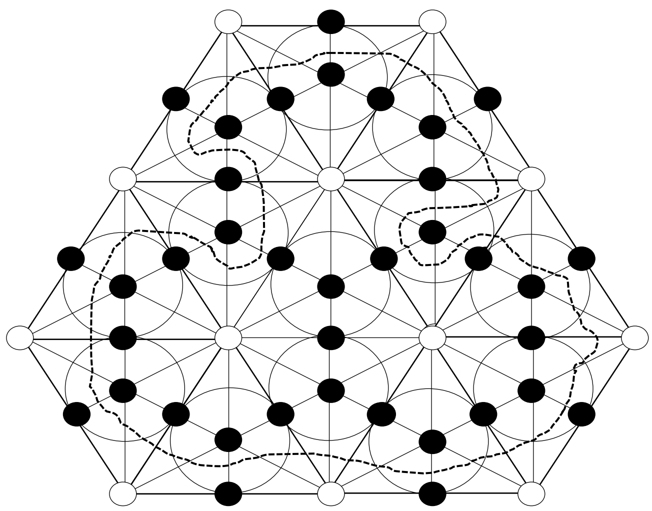

Fig. 6. Example of a healable wound in the proposed 2D- S-(r, t) code

In Fig. 7, another interesting wound example is depicted. It contains 23 nodes and no single node can be regenerated in it. However, if any one of the nodes contained in this would were not corrupted, all nodes could be regenerated.

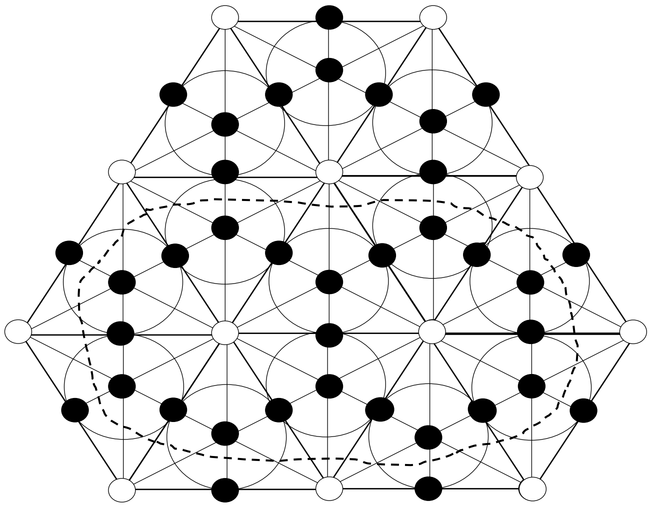

Fig. 7. Illustration of an unhealable wound containing 23 nodes

There could be multiple wounds at the same time and all can be healed if they do not belong to a class of unhealable wounds.

It is illustrated in Fig. 8 where two wounds are depicted. It is interesting that the wound which is on the left side could be healed only after the wound on the right side is healed.

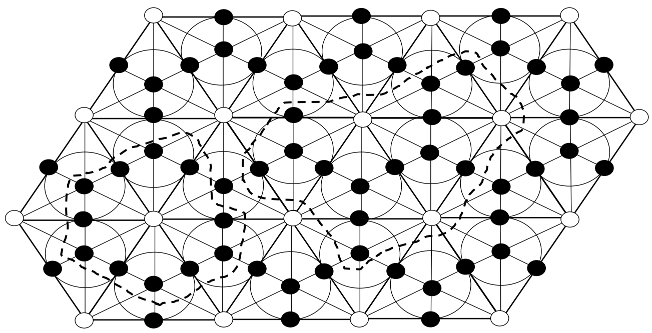

Fig. 8. Illustration of two consecutively healable wounds. After the wound on the right is healed, the left one can also be healed.

These few illustrated examples can be concluded with a statement that further research is needed, probably using computerized simulations to give some more exact or quantitative estimation on wound healing abilities of the proposed codes.

After this rather heuristic discussion and before further research is done, it seems safe to make the conjecture that the proposed family of scalable codes with availability and locality has wound healing abilities which could be useful for disaster recovery.

4 Conclusion

In this paper a generalization of the family of scalable codes from [30] with locality and availability, which could be useful for data regeneration in data centers and for disaster recovery, was presented. Construction of this family of codes could be described as tessellation of a Euclidian two-dimensional plane with graphs of [7, 3, 4] Simplex codes. In this paper it was also pointed out that the new family of the proposed codes has an additional interesting property, namely the ability to regenerate multiple nodes which form a connected graph in some cases. This property is named in this paper as wound healing. Further research is necessary to give some more exact or quantitative estimation on the wound healing abilities of the new codes.

Acknowledgement

This article is based upon work from COST Action 6G-PHYSEC (CA22168), supported by COST (European Cooperation in Science and Technology). This work was also supported by the Slovak Research and Development Agency under Contract No. APVV-19-0436.

References

[1] J. R. Cordeiro and O. Postolache, "Big Data Storage for a Health Predictive System," *2018 International Symposium in Sensing and Instrumentation in IoT Era (ISSI)*, Shanghai, 2018, pp. 1-6. doi: 10.1109/ISSI.2018.8538285

[2] M. Ger, "Understanding Healthcare's New Industry Imperative: Data Chain of Custody," January 27, 2020 Accesed on February 17, 2020, blog.cloudera.com/understanding-healthcares-new-industry imperative-data-chain-of-custody/

[3] S. Ghemawat, H. Gobioff, and S.-T. Leung, "The Google file system," in Proceedings of the 19th ACM Symposium on Operating Systems Principles (SOSP), pages 29–43, 2003

[4] H. Dau, I. Duursma, H. M. Kiah, and O. Milenkovic, "Repairing Reed–Solomon codes with multiple erasures," *IEEE Trans. Inf. Theory*, vol. 64, no. 10, pp. 6567–6582, Oct. 2018

[5] A. G. Dimakis, P. B. Godfrey, Y. Wu, M. J. Wainwright, and K. Ramchandran, "Network coding for distributed storage systems," *IEEE Trans. Inf. Theory*, vol. 56, no. 9, pp. 4539–4551, Sep. 2010.

[6] A. G. Dimakis, K. Ramchandran, Y. Wu and C. Suh, "A Survey on Network Codes for Distributed Storage," in *Proceedings of the IEEE*, vol. 99, no. 3, pp. 476-489, March 2011.

[7] P. Gopalan, C. Huang, H. Simitci, and S. Yekhanin, "On the locality of codeword symbols," *IEEE Trans. Inf. Theory*, vol. 58, no. 11, pp. 6925–6934, Nov. 2011

[8] V. Guruswami and M. Wootters, "Repairing Reed–Solomon codes,"*IEEE Trans. Inf. Theory*, vol. 63, no. 9, pp. 5684–5698, Sep. 2017.

[9] H. Dau and O. Milenkovic, “Optimal repair schemes for some families of full-length Reed–Solomon codes,” in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Jun. 2017, pp. 346–350.

[10] M. Ye and A. Barg, “Explicit constructions of MDS array codes and RS codes with optimal repair bandwidth,” in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Jul. 2016, pp. 1202–1206.

[11] I. Tamo, M. Ye, and A. Barg, “Optimal repair of Reed–Solomon codes: Achieving the cut-set bound,” in *Proc. IEEE 58th Annu. Symp. Found. Comput. Sci. (FOCS)*, Oct. 2017, pp. 216–227.

[12] W. Li, Z. Wang and H. Jafarkhani, "Repairing Reed-Solomon Codes Over GF(2^l)," in *IEEE Communications Letters*, vol. 24, no. 1, pp. 34-37, Jan. 2020.

[13] L. Jin, L. Ma and C. Xing, "Construction of Optimal Locally Repairable Codes via Automorphism Groups of Rational Function Fields," in *IEEE Transactions on Information Theory*, vol. 66, no. 1, pp. 210-221, Jan. 2020

[14] E. Yavari and M. Esmaeili, "Locally Repairable Codes: Joint Sequential–Parallel Repair for Multiple Node Failures," in *IEEE Transactions on Information Theory*, vol. 66, no. 1, pp. 222-232, Jan. 2020.

[15] G. Micheli, "Constructions of Locally Recoverable Codes Which are Optimal," in *IEEE Transactions on Information Theory*, vol. 66, no. 1, pp. 167-175, Jan. 2020.

[16] M. Blaum, "Extended Integrated Interleaved Codes Over Any Field With Applications to Locally Recoverable Codes," in *IEEE Transactions on Information Theory*, vol. 66, no. 2, pp. 936-956, Feb. 2020.

[17] M. Blaum and S. R. Hetzler, "Array Codes with Local Properties," in *IEEE Transactions on Information Theory*, 2020.

[18] X. Xie *et al*., "AZ-Code: An Efficient Availability Zone Level Erasure Code to Provide High Fault Tolerance in Cloud Storage Systems," *2019 35th Symposium on Mass Storage Systems and Technologies (MSST)*, Santa Clara, CA, USA, 2019, pp. 230-243.

[19] P. Huang, E. Yaakobi, H. Uchikawa and P. H. Siegel, "Linear locally repairable codes with availability," *2015 IEEE International Symposium on Information Theory (ISIT)*, Hong Kong, 2015, pp. 1871-1875.

[20] A. Wang, Z. Zhang and M. Liu, "Achieving arbitrary locality and availability in binary codes," *2015 IEEE International Symposium on Information Theory (ISIT)*, Hong Kong, 2015, pp. 1866-1870.

[21] J. Hao, S. Xia and B. Chen, "On the single-parity locally repairable codes with availability," *2016 IEEE/CIC International Conference on Communications in China (ICCC)*, Chengdu, 2016, pp. 1-4.

[22] U. D. Kadiyam, "Locality and Availability with Multiple Erasure Correction," *2018 International Conference on Signal Processing and Communications (SPCOM)*, Bangalore, India, 2018, pp. 16-20.

[23] S. Kruglik, M. Dudina, V. Potapova and A. Frolov, "On one generalization of LRC codes with availability," *2017 IEEE Information Theory Workshop (ITW)*, Kaohsiung, 2017, pp. 26-30.

[24] N. Silberstein, T. Etzion and M. Schwartz, "Locality and Availability of Array Codes Constructed From Subspaces," in *IEEE Transactions on Information Theory*, vol. 65, no. 5, pp. 2648-2660, May 2019.

[25] C. Kim and J. No, "New Constructions of Binary LRCs With Disjoint Repair Groups and Locality 3 Using Existing LRCs," in *IEEE Communications Letters*, vol. 23, no. 3, pp. 406-409, March 2019.

[26] J.-H. Kim, M.-Y. Nam, and H.-Y. Song, “Binary locally repairable codes from complete multipartite graphs,” in *Proc. Int. Conf. ICT Converg. (ICTC)*, Oct. 2015, pp. 1093–1095.

[27] J.-H. Kim, M.-Y. Nam, K.-H. Park, and H.-Y. Song, “Optimal binary locally repairable codes with joint information locality,” in *Proc. IEEE Inf. Theory Workshop (ITW)*, Oct. 2015, pp. 54–58.

[28] J. Kim and H. Song, "Hypergraph-Based Binary Locally Repairable Codes With Availability," in *IEEE Communications Letters*, vol. 21, no. 11, pp. 2332-2335, Nov. 2017.

[29] P. Farkaš, "Scalable Family of Codes with Locality and Availability for Information Repair in Distributed Storage Systems," *2019 4th International Conference on Smart and Sustainable Technologies (SpliTech)*, Split, Croatia, 2019, pp. 1-4.

[30] K. Uehara, Y. R. Chen, M. Hiltunen, K. Joshi and R. Schlichting, "Feasibility Study of Location-Conscious Multi-Site Erasure-Coded Ceph Storage for Disaster Recovery," *2018 IEEE International Conference on Cloud Engineering (IC2E)*, Orlando, FL, 2018, pp. 204-210.

[31] W. Song and C. Yuen, "Binary Locally Repairable Codes - Sequential Repair for Multiple Erasures," *2016 IEEE Global Communications Conference (GLOBECOM)*, Washington, DC, 2016, pp. 1-6.

[32] W. Song, K. Cai, C. Yuen, K. Cai and G. Han, "On Sequential Locally Repairable Codes," in *IEEE Transactions on Information Theory*, vol. 64, no. 5, pp. 3513-3527, May 2018.

[33] E. Yavari and M. Esmaeili, "Locally Repairable Codes: Joint Sequential–Parallel Repair for Multiple Node Failures," in *IEEE Transactions on Information Theory*, vol. 66, no. 1, pp. 222-232, Jan. 2020.

Received 3 January 2024