Cheng, Minquan; Li, Jie; Tang, Xiaohu; Wei, Ruizhong

Linear Coded Caching Scheme for Centralized Networks

Published in:
IEEE Transactions on Information Theory

DOI:
10.1109/TIT.2020.3049074

Published: 01/03/2021

Document Version
Peer reviewed version

Please cite the original version:
Cheng, M., Li, J., Tang, X., & Wei, R. (2021). Linear Coded Caching Scheme for Centralized Networks. *IEEE Transactions on Information Theory*, *67*(3), 1732-1742. [9312631]. https://doi.org/10.1109/TIT.2020.3049074

Linear Coded Caching Scheme for Centralized Networks

Minquan Cheng, Jie Li *Member, IEEE*, Xiaohu Tang *Member, IEEE*, Ruizhong Wei

***Abstract*—Coded caching systems have been widely studied to reduce the data transmission during the peak traffic time. In practice, two important parameters of a coded caching system should be considered, i.e., the transmission rate which is the maximum amount of the data transmission during the peak traffic time, and the subpacketization level, the number of divided packets of each file when we implement a coded caching scheme. Although there exists a tradeoff between transmission rate and subpacketization, we prefer to design a scheme with transmission rate and subpacketization as small as possible since they reflect the transmission efficiency and complexity of the caching scheme, respectively. In this paper, we first characterize a coded caching scheme from the viewpoint of linear algebra and show that designing a linear coded caching scheme is equivalent to constructing three classes of matrices satisfying some rank conditions. Then based on the invariant subspaces in linear algebra and combinatorial design theory, a new class of coded caching schemes over $\mathbb{F}_2$ is obtained by constructing these three classes of matrices. It turns out that the transmission rate of our new scheme is the same as the scheme construct by Yan et al. (IEEE Trans. Inf. Theory 63, 5821-5833, 2017), but the subpacketization is significantly reduced. Finally by means of these matrices, we show that the minimum storage regenerating codes can also be used to construct coded caching schemes.**

***Index Terms*—Linear coded caching scheme, matrices, transmission rate, subpacketization**

I. Introduction

As computer network traffic is increasing at an incredible rate, networks have faced a tremendous pressure on data transmission. Caching is one of the solutions to reduce network loads and has been widely used in heterogeneous wireless networks. The basic idea is simple. During the off peak traffic time, some contents are proactively placed into the user's memory. Clearly if the content is required by the user during the peak traffic time, then traffic amount can be reduced for the peak time. However, for the contents that were not stored into the user's cache, coded caching, which was first proposed by Maddah-Ali and Niesen [11], can also reduce the traffic amount by broadcasting some coded data from the central server and make each user use its cached contents with the broadcasting to recover the desired file.

The work of Cheng was in part supported by NSFC (No.62061004), Guangxi Higher Institutions Program of Introducing 100 High-Level Overseas Talents, Research Fund of Guangxi Key Lab of Multi-source Information Mining & Security (16-B-01), Guangxi Collaborative Innovation Center of Multi-source Information Integration and Intelligent Processing, the Guangxi Bagui Scholar Teams for Innovation and Research Project, and the Guangxi Talent Highland Project of Big Data Intelligence and Application. Li was supported in part by the NSFC (No. 61801176). Tang was supported by NSFC (No. 61871331). Wei was supported by NSERC RGPIN-2016-05610. (Corresponding author: Minquan Cheng.)

M. Cheng is with Guangxi Key Lab of Multi-source Information Mining & Security, Guangxi Normal University, Guilin 541004, China, (e-mail: chengqinshi@hotmail.com).

J. Li is with the Department of Mathematics and Systems Analysis, Aalto University, FI-00076 Aalto, Finland, and also with the Hubei Key Laboratory of Applied Mathematics, Faculty of Mathematics and Statistics, Hubei University, Wuhan 430062, China (e-mail: jie.0.li@aalto.fi, jieli873@gmail.com).

X. Tang is with the Information Security and National Computing Grid Laboratory, Southwest Jiaotong University, Chengdu, 610031, China (e-mail: xhutang@swjtu.edu.cn).

R. Wei is with Department of Computer Science, Lakehead University, Thunder Bay, ON, Canada, P7B 5E1,(e-mail: rwei@lakeheadu.ca).

A. Centralized coded caching

In a centralized (K, M, N) caching system, a single server containing N independent files with the same length connects to K users over a shared link and each user has a cache memory of size M files (see Fig. 1). An F-division (K, M, N) coded caching scheme consists of two phases as follows [11]:

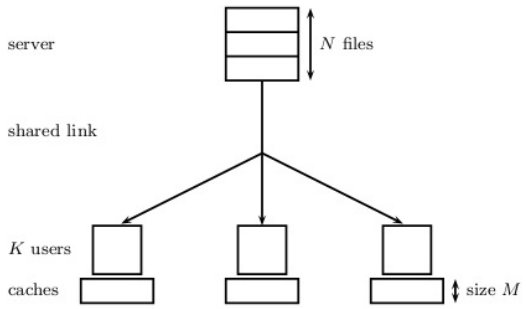

Fig. 1: Caching system

- Placement phase during the off peak traffic times: Each file is divided into F equal packets. Then each user caches some packets (or linear combinations of packets) from the server. If packets are cached directly, it is called uncoded placement; if linear combinations of packets are cached, we call it coded placement. Denote $\mathcal{Z}_k$ the contents cached by user k. In this phase we assume that server does not know the users' requests in the following phase.
- Delivery phase during the peak traffic times: Each user randomly requests one file from N files independently. The server broadcasts a coded signal of size at most R files to users such that each user is able to recover its requested file with the help of its cached contents.

In this paper, R is called the transmission rate of a coded caching scheme. Since the implementation complexity of a coded caching scheme increases along with its subpacketization level, it is desirable to design a scheme with the transmission rate and the subpacketization as small as possible.

B. Prior work

Maddah-Ali and Niesen in [11] proposed the first deterministic scheme. They showed that for any positive integers K, M and N with $M < N$, if KM/N is an integer, there exists a $\binom{K}{KM/N}$-division (K, M, N) coded caching scheme. Such a scheme is referred to as the MN scheme in this paper. In [21], using graph theory the authors showed that when $K \leq N$ the MN scheme has minimum transmission rate under uncoded placement. Hence the MN scheme has also been widely used in various networks [1], [6]–[8], [10]–[13]. Obviously, the subpacketization $F = \binom{K}{KM/N}$ increases rapidly as K increases, which makes this scheme impractical when K is large. So there are many studies on reducing the subpacketization levels. It is well known that there exists a tradeoff between the transmission rate and the subpacketization for a fixed number of users and a fixed memory size.

In this paper we only consider the schemes when $K \leq N$. When $K \leq N$, all of the known discussions on the subpacketizations are proposed under uncoded placement since it is difficult to study the scheme under coded placement. The first relationship between F and R for the caching scheme was discussed by Shanmugam et al. in [15]. That is, when K is large, the authors derived the first tradeoff formula between F and R by probabilistic arguments. Yan et al. in [22] proposed a combinatorial structure called placement delivery array (PDA) to characterize the uncoded placement and delivery strategies. They also showed that MN scheme is equivalent to a special PDA. By constructing PDAs, for any positive integer $q > 1$, the schemes with $\frac{M}{N} = \frac{1}{q}$ and $1 - \frac{1}{q}$ respectively were obtained. It is worth noting that the subpacketization and the transmission rate of the scheme with $\frac{M}{N} = \frac{1}{q}$ achieved the tradeoff formula derived in [15]. There are some other schemes by directly constructing PDAs such as [2], [3] and so on. By PDAs, the authors in [4] showed that given the minimum transmission rate, i.e., the transmission rate of MN scheme, the minimum subpacketization is $F = \binom{K}{KM/N}$, i.e., the subpacketization of MN scheme. We list the schemes achieving the tradeoff of F and R which are from [4], [11], [22], in Table I.

There are many other constructions focusing on further reducing the subpacketization by increasing the transmission rate, such as by the special $(6, 3)$-free hypergraphs [14], the resolvable combinatorial design and linear block codes [19], the (r, t) Ruzsa-Szeméredi graphs [16], [17], the strong edge coloring of bipartite graphs [23], projective space [9] and so on.

C. Main contributions and arrangement of this paper

In this paper, we focus on constructing centralized coded caching schemes with small transmission rate and low subpacketization when $K \leq N$ under coded placement, and obtain the following main results.

- We characterize a coded caching system from the view point of linear algebra and show that a coded caching scheme can be represented by three classes of matrices, say caching matrices, coding matrices and decoding matrices, satisfying some rank conditions.
- By means of linear algebra and combinatorial design theory, the caching matrices, coding matrices and decoding matrices over $\mathbb{F}_2$ are constructed. Consequently a class of deterministic coded caching scheme over $\mathbb{F}_2$ is obtained. Especially for any positive integer q, our scheme has the same user number K, memory ratio $1/q$ and transmission rate $q-1$ as the scheme [22] in the second row of Table I but our subpacketization reduces $q^{\frac{K}{q(q+1)}-1}$ times.
- Based on caching matrices, coding matrices and decoding matrices, we show that the minimum storage regenerating (MSR) codes with optimal repairing bandwidth can be used to construct coded caching schemes. MSR code, which allows distributed storage systems to recover from the loss of a storage node while transmitting the minimum possible amount of data across the network, was proposed by Dimakis et al in [5] and has been a hot topic in the field of distributed storage system recently.

The rest of this paper is organized as follows. The formal statement and linear characterization of coded caching system is introduced in Section II. In Section III, a new characterization of coded caching scheme is proposed from the viewpoint of matrices. In Section IV, a new class of coded caching schemes is proposed. Section V considers the performance of our scheme from the view points of computing complexity, subpacketization and the transmission rate respectively. We indicate that the regenerating code with optimal repair bandwidth can be used to construct a coded caching scheme in Section VI. Finally a conclusion is drawn in Section VII. Some of detailed proofs are attached in Appendixes.

II. Formal problem statement and linear characterization

In this paper, we use bold capital letter, bold lower case letter and curlicue fonts to denote array, vector and set respectively. We denote $[a, b] = \{a, a+1, \ldots, b\}$ and $[a, b) = \{a, a+1, \ldots, b-1\}$ for any integers a and b with $a < b$.

A. Formal problem statement

In the centralized caching system, a server has N files denoted by $\mathcal{W} = \{\mathbf{w}_0, \mathbf{w}_1, \ldots, \mathbf{w}_{N-1}\}$, each of which is a column vector chosen from $\mathbb{F}_p^F$ independently and identically uniformly for some prime power p, and is connected to K users, denoted by $\mathcal{K} = [0, K)$, over a shared link. We assume that each user has a cache memory of size M files and denote the contents in the cache of user $k \in \mathcal{K}$ by $\mathcal{Z}_k$. The formal definition of an F-division (K, M, N) coded caching proposed in [11] as follows.

- Placement phase: Each user k directly accesses the file library $\mathcal{W}$ and stores an arbitrary function thereof in its cache memory, subject to the space limitation of M files. That is, for each user $k \in \mathcal{K}$, there exists a function
$$\phi_k \ : \ \mathbb{F}_p^{NF} \longrightarrow \mathbb{F}_p^{MF}$$
that generates the cache contents $\mathcal{Z}_k \triangleq \phi_k(\mathbf{w}_0, \mathbf{w}_1, \ldots, \mathbf{w}_{N-1})$.
- Delivery phase: Assume that each user requests one file from the files library $\mathcal{W}$ randomly. The request file

TABLE I: MN schemes and PDA schemes from [11], [22]

results	User number K	Memory ratio M/N	Transmission rate R	subpacketization F
MN scheme in [11], K, $t \in \mathbb{Z}^+$ with $t < K$	K	$\frac{t}{K}$	$\frac{K-t}{t+1}$	$\binom{K}{t}$
PDA scheme in [22], m, $q \in \mathbb{Z}^+$ with $q \geq 2$	$(m+1)$q	$\frac{1}{q}$	$q-1$	q^m
PDA scheme in [4], k, $t \in \mathbb{Z}^+$ with $t < k$	$\binom{k}{t+1}$	$1-\frac{t+1}{\binom{k}{t}}$	$\frac{k}{\binom{k}{t}}$	$\binom{k}{t}$
	$\binom{k}{t}$	$\frac{t}{k}$	$\frac{\binom{k}{t+1}}{k}$	k

numbers are denoted by $\mathbf{d} = (d_0, d_1, \cdots, d_{K-1})$, which indicates that user k requests the d_k-th file W_{d_k} for any $d_k \in [0, N)$ and $k \in \mathcal{K}$. The server sends a coded message $\mathbf{x_d}$ with size $R_{\mathbf{d}}$ files to the users such that each user can recover its desired file W_{d_k} from $\mathcal{Z}_k$ and $\mathbf{x_d}$. This phase can be represented by a class of encoding functions and a class of decoding functions:

– The encoding function from the server to user $k \in \mathcal{K}$

$$\psi \ : \ \mathbb{F}_p^{NF} \times \mathbb{F}_p^{KMF} \times [0,N)^K \to \mathbb{F}_p^{R_{\mathbf{d}}F}$$

generates the transmitted message $\mathbf{x_d} \triangleq \psi(\mathcal{W}, \mathcal{Z}_0, \ldots, \mathcal{Z}_{K-1}, \mathbf{d})$ as a function of the library $\mathcal{W}$, the cached information of all users $\{\mathcal{Z}_0, \ldots, \mathcal{Z}_{K-1}\}$ and the demand vector $\mathbf{d}$.

– For each user k, the decoding function

$$\mu_k \ : \ \mathbb{F}_p^{RF} \times \mathbb{F}_p^{MF} \times [0,N)^K \to \mathbb{F}_p^F$$

decodes the request of user k from all messages received by k and its own cache, i.e.,

$$\mathbf{w}_{d_k} = \mu_k(\mathbf{x_d}, \mathcal{Z}_k, \mathbf{d}).$$

R is called the transmission rate of a coded caching scheme.

B. Linear characterization

Assume that the identical caching policy for all files is carried out for each user. If $\frac{FM}{N}$ is a positive integer let us characterize the coded caching scheme when caching, encoding and decoding functions are linear. Since each linear function can be represented by an appropriate matrix, the formal definition of an F-division (K, M, N) linear coded caching scheme introduced in Subsection II-A can be represented by three classes of matrices in the following way.

- In the placement phase, for each $k \in \mathcal{K}$, define

$$\begin{aligned} \mathcal{Z}_k &= \phi_k(\mathbf{w}_0, \mathbf{w}_1, \ldots, \mathbf{w}_{N-1}) \\ &\triangleq (\mathbf{S}_k\mathbf{w}_0, \mathbf{S}_k\mathbf{w}_1, \ldots, \mathbf{S}_k\mathbf{w}_{N-1}) \end{aligned} \tag{1}$$

where $\mathbf{S}_k$, which is called *caching matrix*, is an $\frac{FM}{N} \times F$ matrix over some field $\mathbb{F}_p$. Clearly the size of the contents cached by each user is $\frac{FM}{N}N = MF$.

- In the delivery phase, for any fixed request $\mathbf{d} = (d_0, d_1, \cdots, d_{K-1})$, define

$$\begin{aligned} \mathbf{x_d} &= \psi(\mathcal{W}, \mathcal{Z}_0, \ldots, \mathcal{Z}_{K-1}, \mathbf{d}) \\ &\triangleq \mathbf{A}_0\mathbf{w}_{d_0} + \mathbf{A}_1\mathbf{w}_{d_1} + \cdots + \mathbf{A}_{K-1}\mathbf{w}_{d_K-1} \end{aligned} \tag{2}$$

where $\mathbf{A}_k$, which is called *coding matrix*, is an $RF \times F$ matrix. Then each user k can get the required file

$$\mathbf{w}_{d_k} = \mu_k(\mathbf{x_d}, \mathcal{Z}_k, \mathbf{d}) \triangleq \begin{pmatrix} \mathbf{S}_k\mathbf{w}_{d_k} \\ \mathbf{S}'_k\mathbf{x_d} \end{pmatrix} \tag{3}$$

where $\mathbf{S}'_k$, which is called a *decoding matrix* with order $F(1-\frac{M}{N}) \times RF$. Clearly user k can obtain the required file $\mathbf{w}_{d_k}$ by the contents $\mathcal{Z}_k$ and the message $\mathbf{x_d}$ by (3).

In this paper, a coded caching scheme which can be characterized by the above three classes of matrices, i.e., caching matrices, coding matrices and decoding matrices, is called linear. From the above introductions, we can see that for any coded caching scheme if the uncoded placement strategy satisfying that the identical caching policy for all the files is carried out for each user, then it must be linear. This implies that all the schemes in [2], [3], [9], [14], [16], [17], [19], [22], [23] are linear.

Remark 1: From linear algebra we know that we can get $\mathbf{w}_{d_k}$ if and only if $\mathbf{S}_k\mathbf{w}_k$ and $\mathbf{S}'_k\mathbf{x_d}$ can represent $\mathbf{w}_{d_k}$. So in order to simplify our introduction, we always check whether $\mathbf{S}_k\mathbf{w}_k$ and $\mathbf{S}'_k\mathbf{x_d}$ can represent $\mathbf{w}_{d_k}$ or not in the following.

III. Linear coded caching schemes

From Subsection II-B, we can transfer the linear coded caching problem into an algebra problem. First let us consider the algebraic properties of the caching matrices, coding matrices and decoding matrices for a linear coded caching scheme.

Theorem 1: For any positive integers K, F and Z, if the matrices $\mathbf{S}_k$, $\mathbf{A}_k$ and $\mathbf{S}'_k$, $k \in \mathcal{K}$, defined in (1), (2) and (3) satisfy the following conditions

$$\text{rank}\begin{pmatrix} \mathbf{S}_k \\ \mathbf{S}'_k\mathbf{A}_{k'} \end{pmatrix} = \begin{cases} F, & \text{if } k = k' \\ \frac{FM}{N}, & \text{otherwise} \end{cases} \quad \forall k' \in [0, K), \tag{4}$$

then there exists an F-division (K, M, N) linear coded caching scheme with memory ratio $\frac{M}{N} = \frac{Z}{F}$ and transmission rate R.

Proof. In a caching system, assume that all the files have the same size and can be split into F packets. Then we can take each packet as one symbol in the introduction of the linear coded caching in Subsection II-B. Then we only need to consider the delivery phase. For any request vector $\mathbf{d}$, by (2) the server sends the following coded signal

$$\mathbf{x_d} = \mathbf{A}_0\mathbf{w}_{d_0} + \mathbf{A}_1\mathbf{w}_{d_1} + \cdots + \mathbf{A}_{K-1}\mathbf{w}_{d_K-1}.$$

Each user k can decode the following contents by decoding matrix $\mathbf{S}'_k$

$$\mathbf{S}'_k\mathbf{x_d} = \mathbf{S}'_k\mathbf{A}_0\mathbf{w}_{d_0} + \mathbf{S}'_k\mathbf{A}_1\mathbf{w}_{d_1} + \cdots + \mathbf{S}'_k\mathbf{A}_{K-1}\mathbf{w}_{d_{K-1}}.$$

This is,

$$\mathbf{S}'_k\mathbf{A}_k\mathbf{w}_{d_k} = \mathbf{S}'_k\mathbf{x_d} - \sum_{k'\in[0,K)\setminus\{k\}} \mathbf{S}'_k\mathbf{A}_{k'}\mathbf{w}_{d_{k'}}. \tag{5}$$

Since $\mathrm{rank}\left(\begin{pmatrix}\mathbf{S}_k\\ \mathbf{S}'_k\mathbf{A}_{k'}\end{pmatrix}\right) = \frac{FM}{N}$, then for each $k' \in [0,K)\setminus\{k\}$, $\mathbf{S}'_k\mathbf{A}_{k'}$ can be linear represented by the rows of $\mathbf{S}_k$. Then for each $k' \neq k$, $k' \in [0,K)$, we could get matrix $\mathbf{D}_{k,k'}$ satisfying

$$\mathbf{D}_{k,k'}\mathbf{S}_k = \mathbf{S}'_k\mathbf{A}_{k'}. \tag{6}$$

Submitting (6) into (5), we have

$$\mathbf{S}'_k\mathbf{A}_k\mathbf{w}_{d_k} = \mathbf{S}'_k\mathbf{x_d} - \sum_{k'\in[0,K)\setminus\{k\}} \mathbf{D}_{k,k'}\mathbf{S}_k\mathbf{w}_{d_{k'}}.$$

Then

$$\begin{pmatrix}\mathbf{S}_k\\ \mathbf{S}'_k\mathbf{A}_k\end{pmatrix}\mathbf{w}_{d_k} = \begin{pmatrix}\mathbf{S}_k\mathbf{w}_{d_k}\\ \mathbf{S}'_k\mathbf{A}_k\mathbf{w}_{d_k}\end{pmatrix}$$
$$= \begin{pmatrix}\mathbf{S}_k\mathbf{w}_{d_k}\\ \mathbf{S}'_k\mathbf{x_d} - \sum\limits_{k'\in[0,K)\setminus\{k\}} \mathbf{D}_{k,k'}\mathbf{S}_0\mathbf{w}_{d_{k'}}\end{pmatrix}.$$

Since $\mathrm{rank}\left(\begin{pmatrix}\mathbf{S}_k\\ \mathbf{S}'_k\mathbf{A}_k\end{pmatrix}\right) = F$, user k can decode its request file $\mathbf{w}_{d_k}$, i.e.,

$$\mathbf{w}_{d_k} = \begin{pmatrix}\mathbf{S}_k\\ \mathbf{S}'_k\mathbf{A}_k\end{pmatrix}^{-1} \begin{pmatrix}\mathbf{S}_k\mathbf{w}_{d_k}\\ \mathbf{S}'_k\mathbf{x_d} - \sum\limits_{k'\in[0,K)\setminus\{k\}} \mathbf{D}_{k,k'}\mathbf{S}_0\mathbf{w}_{d_{k'}}\end{pmatrix}. \tag{7}$$

Then the proof is complete. □

From the above proof and the introduction in Subsection II-B, given the caching matrices, coding matrices and decoding matrices satisfying the condition in Theorem 1, an F-division (K, M, N) linear coded caching scheme with $\frac{M}{N} = \frac{Z}{F}$ and R can be obtained by Algorithm 1.

The following example shows how the scheme works.

Example 1: Let $K = N = 6$, $M = 3$ and $F = 4$. We have $\frac{FM}{N} = 2$. For each $k \in [0,6)$, define $\mathbf{S}'_k = \mathbf{S}_k$ and $\mathbf{A}_k$ in the following way.

$$\mathbf{S}_0 = \begin{pmatrix}\mathbf{e}_0\\ \mathbf{e}_2\end{pmatrix}\ \mathbf{S}_1 = \begin{pmatrix}\mathbf{e}_1\\ \mathbf{e}_3\end{pmatrix}\ \mathbf{S}_2 = \begin{pmatrix}\mathbf{e}_0\\ \mathbf{e}_1\end{pmatrix}$$
$$\mathbf{S}_3 = \begin{pmatrix}\mathbf{e}_2\\ \mathbf{e}_3\end{pmatrix}\ \mathbf{S}_4 = \begin{pmatrix}\mathbf{e}_0+\mathbf{e}_1\\ \mathbf{e}_2+\mathbf{e}_3\end{pmatrix}\ \mathbf{S}_5 = \begin{pmatrix}\mathbf{e}_0+\mathbf{e}_2\\ \mathbf{e}_1+\mathbf{e}_3\end{pmatrix}$$
$$\mathbf{A}_0 = \begin{pmatrix}\mathbf{e}_1\\ \mathbf{e}_1\\ \mathbf{e}_3\\ \mathbf{e}_3\end{pmatrix}\ \mathbf{A}_1 = \begin{pmatrix}\mathbf{e}_0\\ \mathbf{e}_0\\ \mathbf{e}_2\\ \mathbf{e}_2\end{pmatrix}\ \mathbf{A}_2 = \begin{pmatrix}\mathbf{e}_2\\ \mathbf{e}_3\\ \mathbf{e}_2\\ \mathbf{e}_3\end{pmatrix}$$
$$\mathbf{A}_3 = \begin{pmatrix}\mathbf{e}_0\\ \mathbf{e}_1\\ \mathbf{e}_0\\ \mathbf{e}_1\end{pmatrix}\ \mathbf{A}_4 = \begin{pmatrix}\mathbf{e}_0\\ 0\\ \mathbf{e}_2\\ 0\end{pmatrix}\ \mathbf{A}_5 = \begin{pmatrix}\mathbf{e}_0\\ \mathbf{e}_1\\ 0\\ 0\end{pmatrix}$$

Algorithm 1 Linear coded caching scheme

procedure PLACEMENT($\mathbf{S}_k$, $\mathcal{W}$)
  Split each file $\mathbf{w}_n \in \mathcal{W}$ into F packets, i.e., $\mathbf{w}_n = (\mathbf{w}_{n,0}, \ldots, \mathbf{w}_{n,F-1})$.
  for $k \in \mathcal{K}$ **do**
    $\mathcal{Z}_k \leftarrow \{\mathbf{S}_k\mathbf{w}_n \mid n \in [0,N)\}$
  end for
end procedure
procedure DELIVERY($\mathbf{A}_k, \mathbf{S}'_k, \mathcal{W}, \mathbf{d}$)
  Server sends $\mathbf{x_d} = \mathbf{A}_0\mathbf{w}_{d_0} + \cdots + \mathbf{A}_{K-1}\mathbf{w}_{d_K-1}$.
  for $k \in \mathcal{K}$ **do**
    for $k' \in \mathcal{K}\setminus\{k\}$ **do**
      User k computes matrix $\mathbf{D}_{k,k'}$ satisfying $\mathbf{D}_{k,k'}\mathbf{S}_k = \mathbf{S}'_k\mathbf{A}_{k'}$.
    end for
    User k gets its required file $\mathbf{w}_{d_k}$ by (7).
  end for
end procedure

Here $\mathbf{e}_j$ is a row vector with length F where the j-th entry is 1 and other entries are 0s, $0 \le j < F$. It is easy to check that (4) holds for each integer $k, k' \in [0,6)$. We assume that $p = 2$. Then we can obtain a 4-division $(6,3,6)$ linear coded caching scheme by Algorithm 1 as follows, where $\mathbf{w}_n = (w_{n,0}, w_{n,1}, w_{n,2}, w_{n,3})^T, w_{n,j} \in \mathbb{F}_2$, $j \in [0,4)$:

- **Placement Phase**: The contents in each users are

$$\mathcal{Z}_0 = \{\mathbf{S}_0\mathbf{w}_n \mid n \in [0,6)\} \quad \mathcal{Z}_1 = \{\mathbf{S}_1\mathbf{w}_n \mid n \in [0,6)\}$$
$$\mathcal{Z}_2 = \{\mathbf{S}_2\mathbf{w}_n \mid n \in [0,6)\} \quad \mathcal{Z}_3 = \{\mathbf{S}_3\mathbf{w}_n \mid n \in [0,6)\}$$
$$\mathcal{Z}_4 = \{\mathbf{S}_4\mathbf{w}_n \mid n \in [0,6)\} \quad \mathcal{Z}_5 = \{\mathbf{S}_5\mathbf{w}_n \mid n \in [0,6)\}$$

- **Delivery Phase**: Assume the request vector $\mathbf{d} = (0, 1, \ldots, 5)$. Then the server just broadcasts

$$\mathbf{x_d} = \mathbf{A}_0\mathbf{w}_0 + \mathbf{A}_1\mathbf{w}_1 + \mathbf{A}_2\mathbf{w}_2 + \mathbf{A}_3\mathbf{w}_3 + \mathbf{A}_4\mathbf{w}_4 + \mathbf{A}_5\mathbf{w}_5.$$

Now let us consider user 0 first. User 0 can obtain the following content by $\mathbf{S}'_0$ and $\mathbf{x_d}$.

$$\begin{aligned}\mathbf{S}'_0\mathbf{x_d} &= \mathbf{S}_0\mathbf{A}_0\mathbf{w}_0 + \mathbf{S}_0\mathbf{A}_1\mathbf{w}_1 + \mathbf{S}_0\mathbf{A}_2\mathbf{w}_2 + \mathbf{S}_0\mathbf{A}_3\mathbf{w}_3\\ &\quad + \mathbf{S}_0\mathbf{A}_4\mathbf{w}_4 + \mathbf{S}_0\mathbf{A}_5\mathbf{w}_5\\ &= \begin{pmatrix}\mathbf{e}_1\\ \mathbf{e}_3\end{pmatrix}\mathbf{w}_0 + \begin{pmatrix}\mathbf{e}_0\\ \mathbf{e}_2\end{pmatrix}\mathbf{w}_1 + \begin{pmatrix}\mathbf{e}_2\\ \mathbf{e}_2\end{pmatrix}\mathbf{w}_2 + \begin{pmatrix}\mathbf{e}_0\\ \mathbf{e}_0\end{pmatrix}\mathbf{w}_3\\ &\quad + \begin{pmatrix}\mathbf{e}_0\\ \mathbf{e}_2\end{pmatrix}\mathbf{w}_4 + \begin{pmatrix}\mathbf{e}_0\\ 0\end{pmatrix}\mathbf{w}_5\end{aligned}$$

That is,

$$\begin{aligned}\begin{pmatrix}\mathbf{e}_1\\ \mathbf{e}_3\end{pmatrix}\mathbf{w}_0 &= \mathbf{S}'_0\mathbf{x_d} - \begin{pmatrix}\mathbf{e}_0\\ \mathbf{e}_2\end{pmatrix}\mathbf{w}_1 - \begin{pmatrix}\mathbf{e}_2\\ \mathbf{e}_2\end{pmatrix}\mathbf{w}_2 - \begin{pmatrix}\mathbf{e}_0\\ \mathbf{e}_0\end{pmatrix}\mathbf{w}_3\\ &\quad - \begin{pmatrix}\mathbf{e}_0\\ \mathbf{e}_2\end{pmatrix}\mathbf{w}_4 - \begin{pmatrix}\mathbf{e}_0\\ 0\end{pmatrix}\mathbf{w}_5.\end{aligned}$$

Then we could get matrices

$$\mathbf{D}_{0,1} = \begin{pmatrix}1&0\\0&1\end{pmatrix}\ \mathbf{D}_{0,2} = \begin{pmatrix}0&1\\0&1\end{pmatrix}\ \mathbf{D}_{0,3} = \begin{pmatrix}1&0\\1&0\end{pmatrix}$$
$$\mathbf{D}_{0,4} = \begin{pmatrix}1&0\\0&1\end{pmatrix}\ \mathbf{D}_{0,5} = \begin{pmatrix}1&0\\0&0\end{pmatrix}$$

satisfying (6). So we have

$$\begin{aligned}\begin{pmatrix}\mathbf{e}_1\\ \mathbf{e}_3\end{pmatrix}\mathbf{w}_0 &= \begin{pmatrix}w_{0,1}\\ w_{0,3}\end{pmatrix}\\ &= \mathbf{S}_0'\mathbf{x_d} - \mathbf{D}_{0,1}\mathbf{S}_0\mathbf{w}_1 - \mathbf{D}_{0,2}\mathbf{S}_0\mathbf{w}_2 - \mathbf{D}_{0,3}\mathbf{S}_0\mathbf{w}_3\\ &\quad -\mathbf{D}_{0,4}\mathbf{S}_0\mathbf{w}_4 - \mathbf{D}_{0,5}\mathbf{S}_0\mathbf{w}_5.\end{aligned}$$

By (7), user 0 can get

$$\begin{aligned}\mathbf{w}_0 &= \begin{pmatrix}\mathbf{S}_0\\ \mathbf{S}_0\mathbf{A}_0\end{pmatrix}^{-1}\begin{pmatrix}w_{0,0}\\ w_{0,2}\\ w_{0,1}\\ w_{0,3}\end{pmatrix}\\ &= \begin{pmatrix}\mathbf{e}_0\\ \mathbf{e}_2\\ \mathbf{e}_1\\ \mathbf{e}_3\end{pmatrix}^{-1}\begin{pmatrix}w_{0,0}\\ w_{0,2}\\ w_{0,1}\\ w_{0,3}\end{pmatrix} = \begin{pmatrix}w_{0,0}\\ w_{0,1}\\ w_{0,2}\\ w_{0,3}\end{pmatrix}.\end{aligned}$$

Similarly we can show that the other users' requests can also be satisfied.

IV. A New construction of linear coded caching scheme

From Theorem 1, we only need to find the three classes of matrices $\mathbf{S}_k$, $\mathbf{A}_k$ and $\mathbf{S}_k'$, $k \in [0, K)$ satisfying the rank conditions in (4). In this section, we propose a concrete construction of these matrices. Throughout this paper we do not specifically distinguish a matrix and the vector space spanned by its rows if the context is clear. Note that (4) holds if and only if the following formula holds.

$$\begin{cases}\mathbf{S}_k'\mathbf{A}_{k'} \subseteq \mathbf{S}_k & \text{if } k \neq k'\\ \mathbf{S}_k + \mathbf{S}_k'\mathbf{A}_{k'} = \mathbb{F}_p^F & \text{if } k = k'\end{cases} \quad k, k' \in [0, K) \tag{8}$$

Here the sum of two subspace $\mathcal{U}$, $\mathcal{V}$ of $\mathbb{F}_p^F$ is defined as $\mathcal{U} + \mathcal{V} = \{\mathbf{u} + \mathbf{v} \mid \mathbf{u} \in \mathcal{U}, \mathbf{v} \in \mathcal{V}\}$.

In the following, we will construct the caching matrices, coding matrices and decoding matrices satisfying (8) over the field $\mathbb{F}_2$. First let us introduce the following subsection.

A. Notations of new constructions

For any positive integers m and $q \geq 2$, we can denote each integer $s \in [0, q^m)$ by $s = (s_0, \ldots, s_{m-1})$ if $s = \sum_{u=0}^{m-1} s_u q^l$ where $0 \leq u < m$ and $s_u \in [0, q)$. We refer to $s = (s_0, \ldots, s_{m-1})_q$ as the *q-ary representation* of s. There are m partitions of $[0, q^m)$, *i.e.*, for each $0 \leq u < m$, the u-th partition is

$$\mathcal{V}_{u,v} = \{s = (s_0, \ldots, s_{m-1}) \mid s_u = v, s \in [0, q^m)\}, \quad 0 \leq v < q. \tag{9}$$

It is easy to check that the following formula holds for any integers u_1, u_2 and v_1, v_2 with $(u_1, v_1) \neq (u_2, v_2)$.

$$\left|\mathcal{V}_{u_1,v_1} \bigcap \mathcal{V}_{u_2,v_2}\right| = \begin{cases}0 & \text{if } u_1 = u_2, v_1 \neq v_2\\ q^{m-2} & \text{if } u_1 \neq u_2\end{cases}$$

In order to construct the caching matrices, the following definitions are needed. Let $\mathbf{e}_s$ be a q^m length row vector where the s-th entry is 1 and other entries are 0s. Define

$$\mathcal{E}_{u,v} = \{\mathbf{e}_s \mid s \in \mathcal{V}_{u,v}\} \tag{10}$$

and

$$\mathcal{Q}_u = \left\{\sum_{s_u=0}^{q-1} \mathbf{e}_{(s_0,\ldots,s_{m-1})} \;\middle|\; s_j \in [0, q), j \neq u\right\} \tag{11}$$

where the sum is performed under modulo q.

Example 2: Let $q = 3$ and $m = 2$. All the integers in $[0, 9)$ can be represented by

$$\begin{aligned}0 = (0,0),\quad 3 = (0,1),\quad 6 = (0,2),\\ 1 = (1,0),\quad 4 = (1,1),\quad 7 = (1,2),\\ 2 = (2,0),\quad 5 = (2,1),\quad 8 = (2,2).\end{aligned}$$

By (9), we have

$$\begin{aligned}\mathcal{V}_{0,0} &= \{(0,0),(0,1),(0,2)\} = \{0,1,2\},\\ \mathcal{V}_{1,0} &= \{(0,0),(1,0),(2,0)\} = \{0,3,6\},\\ \mathcal{V}_{0,1} &= \{(1,0),(1,1),(1,2)\} = \{3,4,5\},\\ \mathcal{V}_{1,1} &= \{(0,1),(1,1),(2,1)\} = \{1,4,7\},\\ \mathcal{V}_{0,2} &= \{(2,0),(2,1),(2,2)\} = \{6,7,8\},\\ \mathcal{V}_{1,2} &= \{(0,2),(1,2),(2,2)\} = \{2,5,8\}.\end{aligned}$$

From (10) and (11) we have

$$\begin{aligned}\mathcal{E}_{0,0} &= \{\mathbf{e}_0,\ \mathbf{e}_3,\ \mathbf{e}_6\}, \quad \mathcal{E}_{0,1} = \{\mathbf{e}_1,\ \mathbf{e}_4,\ \mathbf{e}_7\},\\ \mathcal{E}_{0,2} &= \{\mathbf{e}_2,\ \mathbf{e}_5,\ \mathbf{e}_8\}, \quad \mathcal{E}_{1,0} = \{\mathbf{e}_0,\ \mathbf{e}_1,\ \mathbf{e}_2\},\\ \mathcal{E}_{1,1} &= \{\mathbf{e}_3,\ \mathbf{e}_4,\ \mathbf{e}_5\}, \quad \mathcal{E}_{1,2} = \{\mathbf{e}_6,\ \mathbf{e}_7,\ \mathbf{e}_8\}\end{aligned}$$

and

$$\begin{aligned}\mathcal{Q}_0 &= \{\mathbf{e}_0 + \mathbf{e}_1 + \mathbf{e}_2,\ \mathbf{e}_3 + \mathbf{e}_4 + \mathbf{e}_5,\ \mathbf{e}_6 + \mathbf{e}_7 + \mathbf{e}_8\},\\ \mathcal{Q}_1 &= \{\mathbf{e}_0 + \mathbf{e}_3 + \mathbf{e}_6,\ \mathbf{e}_1 + \mathbf{e}_4 + \mathbf{e}_7,\ \mathbf{e}_2 + \mathbf{e}_5 + \mathbf{e}_8\}.\end{aligned}$$

Now let us introduce the essential functions to construct coding matrices. For each $u \in [0, m)$, $v \in [0, q)$ and for each $s \in [0, q^m)$, let $\varphi_{u,v}(s)$ be the vector where the u-th entry of s is v and the other entries are same as that of s, and $\phi_{u,v}(s) = 1$ if $s \in \mathcal{V}_{u,v}$ otherwise 0. Using the above functions, for any two distinct $v, v' \in [0, q)$ we can define a binary $q^m \times q^m$ matrix

$$\begin{aligned}\mathbf{C}_{u;v,v'} = \big(&\phi_{u,v'}(0)(\mathbf{e}_{\varphi_{u,v}(0)} + \mathbf{e}_0)^T, \ldots,\\ &\phi_{u,v'}(q^m - 1)(\mathbf{e}_{\varphi_{u,v}(q^m-1)} + \mathbf{e}_{q^m-1})^T\big)\end{aligned} \tag{12}$$

i.e., the s-th column is the sum of $\mathbf{e}_{\varphi_{u,v}(s)}^T$ and $\mathbf{e}_s^T$ if $s \in \mathcal{V}_{u,v'}$, and others are zero columns, and a binary $q^m \times q^m$ matrix

$$\mathbf{C}_{u;q,v} = \big(\phi_{u,v}(0)\mathbf{e}_0^T, \ldots, \phi_{u,v}(q^m - 1)\mathbf{e}_{q^m-1}^T\big) \tag{13}$$

i.e., the s-th column is $\mathbf{e}_s^T$ if $s \in \mathcal{V}_{u,v}$, and the others are zero columns.

Example 3: When $q = 3$ and $m = 2$, from Example 2 we have

$$\begin{aligned}
&\mathbf{C}_{0,0,1} = (0, \mathbf{e}_0^T + \mathbf{e}_1^T, 0, 0, \mathbf{e}_3^T + \mathbf{e}_4^T, 0, 0, \mathbf{e}_6^T + \mathbf{e}_7^T, 0),\\
&\mathbf{C}_{0,0,2} = (0, 0, \mathbf{e}_0^T + \mathbf{e}_2^T, 0, 0, \mathbf{e}_3^T + \mathbf{e}_5^T, 0, 0, \mathbf{e}_6^T + \mathbf{e}_8^T),\\
&\mathbf{C}_{0,1,0} = (\mathbf{e}_0^T + \mathbf{e}_1^T, 0, 0, \mathbf{e}_3^T + \mathbf{e}_4^T, 0, 0, \mathbf{e}_6^T + \mathbf{e}_7^T, 0, 0),\\
&\mathbf{C}_{0,1,2} = (0, 0, \mathbf{e}_1^T + \mathbf{e}_2^T, 0, 0, \mathbf{e}_4^T + \mathbf{e}_5^T, 0, 0, \mathbf{e}_7^T + \mathbf{e}_8^T),\\
&\mathbf{C}_{0,2,0} = (\mathbf{e}_0^T + \mathbf{e}_2^T, 0, 0, \mathbf{e}_3^T + \mathbf{e}_5^T, 0, 0, \mathbf{e}_6^T + \mathbf{e}_8^T, 0, 0),\\
&\mathbf{C}_{0,2,1} = (0, \mathbf{e}_1^T + \mathbf{e}_2^T, 0, 0, \mathbf{e}_4^T + \mathbf{e}_5^T, 0, 0, \mathbf{e}_7^T + \mathbf{e}_8^T, 0),\\
&\mathbf{C}_{1,0,1} = (0, 0, 0, \mathbf{e}_0^T + \mathbf{e}_3^T, \mathbf{e}_1^T + \mathbf{e}_4^T, \mathbf{e}_2^T + \mathbf{e}_5^T, 0, 0, 0),\\
&\mathbf{C}_{1,0,2} = (0, 0, 0, 0, 0, 0, \mathbf{e}_0^T + \mathbf{e}_6^T, \mathbf{e}_1^T + \mathbf{e}_7^T, \mathbf{e}_2^T + \mathbf{e}_8^T),\\
&\mathbf{C}_{1,1,0} = (\mathbf{e}_0^T + \mathbf{e}_3^T, \mathbf{e}_1^T + \mathbf{e}_4^T, \mathbf{e}_2^T + \mathbf{e}_5^T, 0, 0, 0, 0, 0, 0),\\
&\mathbf{C}_{1,1,2} = (0, 0, 0, 0, 0, 0, \mathbf{e}_3^T + \mathbf{e}_6^T, \mathbf{e}_4^T + \mathbf{e}_7^T, \mathbf{e}_5^T + \mathbf{e}_8^T),\\
&\mathbf{C}_{0,2,0} = (\mathbf{e}_0^T + \mathbf{e}_6^T, \mathbf{e}_1^T + \mathbf{e}_7^T, \mathbf{e}_2^T + \mathbf{e}_8^T, 0, 0, 0, 0, 0, 0),\\
&\mathbf{C}_{0,2,1} = (0, 0, 0, \mathbf{e}_3^T + \mathbf{e}_6^T, \mathbf{e}_4^T + \mathbf{e}_7^T, \mathbf{e}_5^T + \mathbf{e}_8^T, 0, 0, 0)
\end{aligned}$$

by (12) and

$$\begin{aligned}
\mathbf{C}_{1,3,0} &= (\mathbf{e}_0^T, \mathbf{e}_1^T, \mathbf{e}_2^T, 0, 0, 0, 0, 0, 0),\\
\mathbf{C}_{1,3,1} &= (0, 0, 0, \mathbf{e}_3^T, \mathbf{e}_4^T, \mathbf{e}_5^T, 0, 0, 0)
\end{aligned}$$

by (13).

From the above notations, the following useful result can be obtained.

Lemma 1: Given positive integers m and $q \geq 2$, sets and matrices in (10), (11), (12) and (13) satisfy the following conditions: For any integers u_1, $u_2 \in [0, m)$ and any integers v_1, $v_2 \in [0, q)$, $v_3 \in [0, q]$,

$$\mathcal{E}_{u_1,v_1}\mathbf{C}_{u_2,v_3,v_2} = \begin{cases} \mathcal{E}_{u_2,v_2} & \text{if } \begin{matrix} u_1 = u_2 \text{ and} \\ v_1 = v_2 (\text{ or } v_1 = v_3) \end{matrix} \\ \mathcal{E}_{u_1,v_1} \bigcap \mathcal{E}_{u_2,v_2} & \text{otherwise} \end{cases} \tag{14}$$

$$\mathcal{Q}_{u_1}\mathbf{C}_{u_2,v_3,v_2} = \begin{cases} \mathcal{E}_{u_2,v_2} & \text{if } u_1 = u_2, v_3 = q \\ \{0\} & \text{if } u_1 = u_2, v_3 < q \\ \mathcal{Q}_{u_1} & \text{otherwise} \end{cases} \tag{15}$$

The proof of Lemma 1 is included in Appendix A. For any positive integer z with $1 \leq z < q$, we can define $\lfloor \frac{q-1}{q-z} \rfloor$ sets

$$\mathcal{G}_{v,\varepsilon} = \{v + 1 + \varepsilon(q-z), \ldots, v + (q-z) + \varepsilon(q-z)\}_q \tag{16}$$

where $v \in [0, q)$ and $\varepsilon \in [0, \lfloor \frac{q-1}{q-z} \rfloor)$. Here $\mathcal{A}_q = \{a \pmod q \mid a \in \mathcal{A}\}$ for any given integer set $\mathcal{A}$. For example when $q = 3$ and $z = 2$, we have $\lfloor \frac{q-1}{q-z} \rfloor = 1$. By (16), we have

$$\mathcal{G}_{0,0} = \{1, 2\}, \quad \mathcal{G}_{1,0} = \{2, 0\}, \quad \mathcal{G}_{2,0} = \{0, 1\}. \tag{17}$$

B. New construction

Construction 1: Given positive integers $m \geq 1$, $q \geq 2$ and z with $z < q$, each user k is represented by tuples (u, v, ε) or (u, q, ε) in the following for convenience, where $0 \leq u < m$, $0 \leq v < q$ and $0 \leq \varepsilon < \lfloor \frac{q-1}{q-z} \rfloor$. From (12), (13) and (16), for any (u, v, ε) and (u, q, ε) we can construct the caching matrices as follows

$$\begin{aligned}
\mathbf{S}_{u,v,\varepsilon} &= \{\mathcal{E}_{u,v'} \mid v' \in [0, q) \setminus G_{v,\varepsilon}\},\\
\mathbf{S}_{u,q,\varepsilon} &= \mathcal{Q}_u \bigcup \{\mathcal{E}_{u,v'} \mid v' \in [0, q-1) \setminus G_{q-1,\varepsilon}\}.
\end{aligned} \tag{18}$$

The coding matrices are

$$\begin{aligned}
\mathbf{A}_{u,v,\varepsilon} &= \begin{pmatrix} \mathbf{C}_{u,v,v_1} \\ \mathbf{C}_{u,v,v_2} \\ \vdots \\ \mathbf{C}_{u,v,v_{q-z}} \end{pmatrix}, v_i \in \mathcal{G}_{v,\varepsilon}, \quad \text{and}\\
\mathbf{A}_{u,q,\varepsilon} &= \begin{pmatrix} \mathbf{C}_{u,q,v'_1} \\ \mathbf{C}_{u,q,v'_2} \\ \vdots \\ \mathbf{C}_{u,q,v'_{q-z}} \end{pmatrix}, v'_i \in \mathcal{G}_{q-1,\varepsilon}.
\end{aligned} \tag{19}$$

And the decoding matrices are

$$\begin{aligned}
\mathbf{S}'_{u,v,\varepsilon} &= diag(\underbrace{\mathcal{E}_{u,v}, \ldots, \mathcal{E}_{u,v}}_{q-z}),\\
\mathbf{S}'_{u,q,\varepsilon} &= diag(\underbrace{\mathcal{Q}_u, \ldots, \mathcal{Q}_u}_{q-z}).
\end{aligned} \tag{20}$$

Based on Construction 1, the following result can be obtained.

Theorem 2: For any positive integers q, z, m with $q \geq 2$ and $z < q$, there exists a q^m-division (K, M, N) linear coded caching scheme with parameters $K = m(q+1)\lfloor \frac{q-1}{q-z} \rfloor$, memory ratio $\frac{M}{N} = \frac{z}{q}$ and transmission rate $R = q - z$. The operation is over the finite field $\mathbb{F}_2$.

The proof of Theorem 2 is included in Appendix B.

Example 4: When $q = 3$, $m = 2$, we can obtain the matrices $\mathcal{E}_{u_1,v_1}$, $\mathcal{Q}_{u_1}$ and C_{u_1,v_2,v_1} from Example 2 and Example 3 for each $u_1 \in [0, 2)$, $v_1 \in [0, 3)$, $v_2 \in [0, 3]$. If $z = 1$, we have $\lfloor \frac{q-1}{q-z} \rfloor = 1$. Then we have caching matrices

$$\begin{aligned}
&\mathbf{S}_{0,0,0} = \mathcal{E}_{0,0} \quad \mathbf{S}_{0,1,0} = \mathcal{E}_{0,1} \quad \mathbf{S}_{0,2,0} = \mathcal{E}_{0,2} \quad \mathbf{S}_{1,0,0} = \mathcal{E}_{1,0}\\
&\mathbf{S}_{1,1,0} = \mathcal{E}_{1,1} \quad \mathbf{S}_{1,2,0} = \mathbf{Q}_{1,2} \quad \mathbf{S}_{0,3,0} = \mathcal{Q}_{0,3} \quad \mathbf{S}_{1,3,0} = \mathcal{Q}_{1,3}
\end{aligned}$$

by (17) and (18), coding matrices

$$\begin{aligned}
&\mathbf{A}_{0,0,0} = \begin{pmatrix} \mathbf{C}_{0,0,1} \\ \mathbf{C}_{0,0,2} \end{pmatrix} \quad \mathbf{A}_{0,1,0} = \begin{pmatrix} \mathbf{C}_{0,1,0} \\ \mathbf{C}_{0,1,2} \end{pmatrix} \quad \mathbf{A}_{0,2,0} = \begin{pmatrix} \mathbf{C}_{0,2,0} \\ \mathbf{C}_{0,2,1} \end{pmatrix}\\
&\mathbf{A}_{1,0,0} = \begin{pmatrix} \mathbf{C}_{1,0,1} \\ \mathbf{C}_{1,0,2} \end{pmatrix} \quad \mathbf{A}_{1,1,0} = \begin{pmatrix} \mathbf{C}_{1,1,0} \\ \mathbf{C}_{1,1,2} \end{pmatrix} \quad \mathbf{A}_{1,2,0} = \begin{pmatrix} \mathbf{C}_{0,2,0} \\ \mathbf{C}_{0,2,1} \end{pmatrix}\\
&\mathbf{A}_{0,3,0} = \begin{pmatrix} \mathbf{C}_{0,3,0} \\ \mathbf{C}_{0,3,1} \end{pmatrix} \quad \mathbf{A}_{1,3,0} = \begin{pmatrix} \mathbf{C}_{1,3,0} \\ \mathbf{C}_{1,3,1} \end{pmatrix}
\end{aligned}$$

by (17) and (19), and decoding matrices

$$\begin{aligned}
&\mathbf{S}'_{0,0,0} = \begin{pmatrix} \mathcal{E}_{0,0} & 0 \\ 0 & \mathcal{E}_{0,0} \end{pmatrix} \quad \mathbf{S}'_{0,1,0} = \begin{pmatrix} \mathcal{E}_{0,1} & 0 \\ 0 & \mathcal{E}_{0,1} \end{pmatrix}\\
&\mathbf{S}'_{0,2,0} = \begin{pmatrix} \mathcal{E}_{0,2} & 0 \\ 0 & \mathcal{E}_{0,2} \end{pmatrix} \quad \mathbf{S}'_{1,0,0} = \begin{pmatrix} \mathcal{E}_{1,0} & 0 \\ 0 & \mathcal{E}_{1,0} \end{pmatrix}\\
&\mathbf{S}'_{1,1,0} = \begin{pmatrix} \mathcal{E}_{1,1} & 0 \\ 0 & \mathcal{E}_{1,1} \end{pmatrix} \quad \mathbf{S}_{1,2,0} = \begin{pmatrix} \mathcal{E}_{1,2} & 0 \\ 0 & \mathcal{E}_{1,2} \end{pmatrix}\\
&\mathbf{S}'_{0,3,0} = \begin{pmatrix} \mathcal{Q}_0 & 0 \\ 0 & \mathcal{Q}_0 \end{pmatrix} \quad \mathbf{S}_{1,3,0} = \begin{pmatrix} \mathcal{Q}_1 & 0 \\ 0 & \mathcal{Q}_1 \end{pmatrix}
\end{aligned}$$

by (17) and (20). We can check that the above caching matrices, encoding matrices and decoding matrices satisfy (8). So we can obtain an 8-division $(8, M, N)$ linear coded caching scheme with $M/N = 1/3$ and transmission rate $R = 2$.

V. Performance analyses

From Table I, we only need to consider the performance of our new scheme by comparing with the schemes from [11], [22] since they have minimum transmission rate or approximating minimum transmission. From Theorem 2 we know that the computed field of our new scheme has the same computed field of the scheme from [11], [22]. Now let us consider the computing complexity, transmission rate and the subpacketization of our scheme respectively.

A. Low computing complexity

Let us consider the caching matrices, coding matrices and decoding matrices respectively in Construction 1. First let us consider the caching matrices in (18) in the placement phase. Clearly each caching matrix $\mathbf{S}_{u,v,\varepsilon}$ is composed by $\mathcal{E}_{u,v}$ where $u \in [0,m)$, $v,v' \in [0,q)$ and $\varepsilon \in [0, \lfloor \frac{q-1}{q-z} \rfloor)$. There are $mq\lfloor \frac{q-1}{q-z} \rfloor$ such matrices. By (10) and (1), all the caching strategies for the users which are represented by (u,v,ε) are uncoded placement. Each matrix $\mathbf{S}_{u,q,\varepsilon}$ is composed by $\mathcal{Q}_u$ and $\mathcal{E}_{u,v}$. There are exact $m\lfloor \frac{q-1}{q-z} \rfloor$ such matrices. Since $\mathcal{Q}_u$ consists of q^{m-1} vectors, each of them has exactly q ones, each matrix $\mathbf{S}_{u,q,\varepsilon}$ has exactly q^{m-1} rows with q entries of 1 and other entries of 0 and $q^m - q^{m-1}$ row with one entry of 1 and the other entries of 0. So there are only q^{m-1} coded packets each of which is the sum of q packets and left packets are directly cached for the users which are represented by (u,q,ε). So our scheme just has extra $m\lfloor \frac{q-1}{q-z} \rfloor (q-1)q^{m-1}$ additions compared with the uncoded placement in the placement phase.

In the delivery phase, let us consider the encoding matrices in (19). By (12) and (13) each row of encoding matrix has at most one entry of 1. By (20) there are exact $mq\lfloor \frac{q-1}{q-z} \rfloor$ decoding matrices, where each row has one entry of 1 and the other entries of 0, and $m\lfloor \frac{q-1}{q-z} \rfloor$ decoding matrices, where each row has q entries of 1 and the other entries of 0.

From the above discussions, we see that compared to the scheme under uncoded caching placement, our new scheme adds some additional computation. However the additional computing is very limited.

B. Small transmission rate and low subpacketization

For any positive integers K and $q \geq 2$, assume that there exist two positive integers m and m' such that $K = (m+1)q = (q+1)m'$. Let $t = m+1$. From the first row of Table I, we have a (K,M,N) MN scheme with memory ratio, transmission rate and the subpacketization as follows.

$$\frac{M}{N} = \frac{1}{q} \quad R_{MN} = \frac{K}{q+K}(q-1) \quad F_{MN} = \binom{K}{K/q} \quad (21)$$

From the second row of Table I, we have a PDA scheme with memory ratio, transmission rate and the subpacketization as follows.

$$\frac{M}{N} = \frac{1}{q} \quad R_{PDA} = q-1 \quad F_{PDA} = q^m = q^{\frac{K}{q}-1} \quad (22)$$

From Theorem 2, we have a linear coded caching scheme with memory ratio, transmission rate and the subpacketization as follows.

$$\frac{M}{N} = \frac{1}{q} \quad R_{Linear} = q-1 \quad F_{Linear} = q^{m'} = q^{\frac{K}{q+1}} \quad (23)$$

It is not difficult to check that when K is large, the transmission rate R_{PDA} approximates R_{MN} and the subpacketization F_{PDA} is far smaller than the subpacketization F_{MN}. This investigations have been proved by Yan et al., in [22]. By (21) and (23), the transmission rate of our scheme approximates the transmission rate of MN scheme when K is large. Compared with PDA scheme, which achieves the tradeoff formula derived in [15] between transmission rate and subpacketization, with the same user number, memory ratio and transmission rate, the ratio

$$\frac{F_{PDA}}{F_{Linear}} = \frac{q^{\frac{K}{q}-1}}{q^{\frac{K}{q+1}}} = q^{\frac{K}{q(q+1)}-1}.$$

Clearly when K is large, the ratio $\frac{F_{PDA}}{F_{Linear}}$ approaches infinity. This implies that when K is large, the subpacketizations of our scheme is much smaller while the transmission rate does not increase.

Example 5: Assume that $\frac{M}{N} = \frac{1}{2}$. The MN scheme, PDA scheme from Table I and the scheme from Theorem 2 can be obtained in Table II. Clearly when K is larger, the advantage on the subpacketization of our scheme is more obvious.

TABLE II: MN schemes, PDA schemes, and the new schemes when $\frac{M}{N} = \frac{1}{2}$

K	$\frac{M}{N}$	MN schemes [11]		PDA schemes [22]		Linear schemes	
		R	F	R	F	R	F
12	0.5	0.8571	924	1	32	1	16
18	0.5	0.9	48620	1	256	1	64
24	0.5	0.9231	2704156	1	2048	1	256
30	0.5	0.9375	155117520	1	16384	1	1024
36	0.5	0.9474	9075135300	1	131072	1	4096

VI. Constructions from minimum storage regenerating codes

In this section, an interesting relationship between the linear coded caching scheme and the Minimum storage regenerating (MSR) code introduced in [5] for distributed storage systems is proposed. Consequently all the MSR codes with optimal repairing bandwidth can be used to directly construct linear coded caching schemes.

Firstly let us briefly introduce the module and requirements of MSR codes with optimal repairing bandwidth. Assume that a file of length KF denoted by a column vector of $\mathbf{F}_p^{KF}$ is partitioned into K parts $\mathbf{w}_0, \mathbf{w}_1, \cdots, \mathbf{w}_{K-1}$ each of size F, where p is a prime power. We encode this file using an $(n = K + r, K)$ MSR code $\mathcal{C}$

$$\mathcal{C} = \begin{pmatrix} \mathbf{A}_{0,0} & \cdots & \mathbf{A}_{0,K-1} \\ \vdots & \ddots & \vdots \\ \mathbf{A}_{r-1,0} & \cdots & \mathbf{A}_{r-1,K-1} \end{pmatrix}. \quad (24)$$

as listed in Table III. Here $\mathbf{A}_{i,k}$, which is a nonsingular $F \times F$ matrix, is called the *encoding matrix*. It is worth noting that the nonsingular property of encoding matrix is necessary to guarantee the resiliency to any $n-K$ node failures. This is also called maximum distance separable (MDS) property. Precisely, the first K (systematic) nodes store subfiles $\mathbf{w}_0, \mathbf{w}_1,$

TABLE III: Structure of a $(K+r, K)$ MSR code

Systematic node	Systematic datas
0	$\mathbf{w}_0$
$\vdots$	$\vdots$
$K-1$	$\mathbf{w}_{K-1}$
Parity node	Parity datas
K	$\mathbf{w}_K = \mathbf{A}_{0,0}\mathbf{w}_0 + \cdots + \mathbf{A}_{0,K-1}\mathbf{w}_{K-1}$
$K+1$	$\mathbf{w}_{K+1} = \mathbf{A}_{1,0}\mathbf{w}_0 + \cdots + \mathbf{A}_{1,K-1}\mathbf{w}_{K-1}$
$\vdots$	$\vdots$
$K+r-1$	$\mathbf{w}_{K+r-1} = \mathbf{A}_{r-1,0}\mathbf{w}_0 + \cdots + \mathbf{A}_{r-1,K-1}\mathbf{w}_{K-1}$

$\ldots$, $\mathbf{w}_{K-1}$ respectively, and the parity node $K+i$, $i \in [0, r)$, stores $\mathbf{w}_{K+i} = \sum_{k=0}^{K-1} \mathbf{A}_{i,k}\mathbf{w}_k$.

Lemma 2: ([18]) When one systematic node fails, the code defined in (24) has optimal repairing bandwidth if there exist matrices $\mathbf{S}_{i,k}$ each of dimension F/r, such that for any k, $k' \in [0, K)$ and $i \in [0, r)$

$$\text{rank}\begin{pmatrix} \mathbf{S}_{0,k}\mathbf{A}_{0,k'} \\ \vdots \\ \mathbf{S}_{r-1,k}\mathbf{A}_{r-1,k'} \end{pmatrix} = \begin{cases} F, & \text{if } k = k' \\ \frac{F}{r}, & \text{otherwise} \end{cases} \tag{25}$$

Without loss of generality [18], we can assume that $\mathbf{A}_{0,0}$, $\ldots$, $\mathbf{A}_{0,K-1}$, are identity matrices i.e., $\mathbf{A}_{0,0} = \ldots = \mathbf{A}_{0,K-1} = I$. From the aforementioned discussion, the following relationship between coded caching scheme and minimum storage regenerating code can be obtained.

Theorem 3: Given an $(n = K + r, K)$ MSR code $\mathcal{C}$ defined in (24) with optimal repairing bandwidth and the size of systematic node F, an F-division (K, M, N) linear caching scheme with memory ratio $\frac{M}{N} = \frac{1}{r}$ and transmission rate $R = r - 1$ can be obtained.

Proof. From Lemma 2, there exist matrices $\mathbf{S}_{i,k}$ each of dimension $\frac{F}{r}$, such that formula (25) holds for any k, $k' \in [0, K)$ and $i \in [0, r)$. For each $k \in [0, K)$, we can define

$$\mathbf{S}_k = \mathbf{S}_{0,k}\mathbf{A}_{0,k} = \mathbf{S}_{0,k},\ \mathbf{S}'_k = \begin{pmatrix} \mathbf{S}_{1,k} & & \\ & \ddots & \\ & & \mathbf{S}_{r-1,k} \end{pmatrix}$$

$$\text{and } \mathbf{A}_k = \begin{pmatrix} \mathbf{A}_{1,k} \\ \vdots \\ \mathbf{A}_{r-1,k} \end{pmatrix}.$$

Clearly, for any k, $k' \in [k]$, we have

$$\begin{aligned} \text{rank}\begin{pmatrix} \mathbf{S}_k \\ \mathbf{S}'_k\mathbf{A}_{k'} \end{pmatrix} &= \text{rank}\begin{pmatrix} \mathbf{S}_{0,k}\mathbf{A}_{0,k'} \\ \vdots \\ \mathbf{S}_{r-1,k}\mathbf{A}_{r-1,k'} \end{pmatrix} \\ &= \begin{cases} F, & \text{if } k = k' \\ \frac{F}{r}, & \text{otherwise} \end{cases} \end{aligned}$$

Clearly the transmission rate $R = \frac{(r-1)F}{F} = r - 1$. That completes the proof. □

Clearly when we construct a linear coded caching scheme by means of an MSR code from Theorem 3, the number of systematic nodes is as large as possible. Wang et al., in [20] proposed an MSR code with optimal repair bandwidth such that the number of systematic nodes K is the largest.

Lemma 3: ([20]) There exists an $(n = q + K, K = (q+1)m)$ MSR code with optimal repair bandwidth and the size of a node $F = q^m$ over $\mathbb{F}_p$ for any positive integer m and q, where p is a prime power and larger than q.

From Theorem 3 and Lemma 3, the following result can be obtained.

Corollary 1: For any positive integers m and $q \geq 2$, there exists a q^m-division $((q+1)m, M, N)$ linear coded caching scheme with $\frac{M}{N} = \frac{1}{q}$ and $R = q - 1$ over $\mathbb{F}_p$, where $p > q$ is a prime power.

Clearly for any positive integers $q \geq 2$ and m, when $z = 1$ the scheme from Theorem 2 and the scheme from Corollary 1 have all the same parameters. However the scheme from Theorem 2 is over $\mathbb{F}_2$, while the scheme from Corollary 1 is over $\mathbb{F}_p$, where $p > q$ is a prime power. In fact when constructing coding matrices $\mathbf{A}_k$ in Theorem 3, the MDS property is not necessary. It is interesting to know if we can ignore the MDS property of the MSR code to get better results. So we have the following question.

Open problem: How to construct linear coded caching schemes by modifying the constructions of MSR codes for reducing the computed field or getting more classes of schemes.

Even though our work is completed independently of the work in [20], the used key sets in (10) and (11) are the same. In [20], the authors first set that the encoding matrices $\mathbf{A}_{0,k}$, $0 \leq k < K$, equals identity matrix I. Then they constructed each other encoding matrix, say $\mathbf{A}_{i,k}$, by choosing appropriate q elements from set $\{\mathcal{E}_{u,v}, \mathcal{Q}_u \mid 0 \leq u < m, 0 \leq v < q\}$ as eigenvectors and choosing related q distinct non-zero elements from $\mathbb{F}_p$ as eigenvalues. For the details the interested reader could be referred to [20]. However this method is not suitable for further reducing the computed field when we construct a linear coded caching scheme.

VII. Conclusion

In this paper, we first characterized a linear coded caching scheme using linear algebra, which generalized a majority of previous constructions. Consequently, the problem of designing a linear coded caching scheme is equivalent to constructing three classes of matrices satisfying some rank conditions. Then by constructing these three classes of matrices, a new linear coded caching scheme over $\mathbb{F}_2$ was obtained. Compared with the scheme construct by Yan et al. in [22], our new scheme has the same user number, memory ratio and transmission rate but has much smaller supbacketization. Finally by means of these three classes of matrices, we proved that the optimal minimum storage regenerating codes can be used to construct coded caching schemes.

Appendix A: Proof of Lemma 1

Proof. Let us consider (14) first. According to the values of u_1 and u_2, we only need to consider the following cases.

- When $u_1 = u_2$, let us consider the value of v_3

– If $v_3 < q$, we have

$$\begin{aligned}
&\mathcal{E}_{u_1,v_1}\mathbf{C}_{u_1,v_3,v_2}\\
&=\Big\{\mathbf{e}_s\left(\phi_{u_1,v_2}(s')(\mathbf{e}_{\varphi_{u_1,v_3}(s')}+\mathbf{e}_{s'})^T\right)_{s'\in[0,q^m)}\\
&\qquad\qquad\Big|\ s\in\mathcal{E}_{u_1,v_1}\Big\}\\
&=\Big\{\left(\phi_{u_1,v_2}(s')(\mathbf{e}_s\mathbf{e}^T_{\varphi_{u_1,v_3}(s')}+\mathbf{e}_s\mathbf{e}^T_{s'})\right)_{s'\in[0,q^m)} \quad (26)\\
&\qquad\qquad\Big|\ s\in\mathcal{E}_{u_1,v_1}\Big\}\\
&=\Big\{\mathbf{e}_{\varphi_{u_1,v_2}(s)}\mid s\in\mathcal{E}_{u_1,v_1}\bigcap\mathcal{E}_{u_1,v_3}\Big\}\\
&\quad\bigcup\left(\mathcal{E}_{u_1,v_1}\bigcap\mathcal{E}_{u_1,v_2}\right)
\end{aligned}$$

Here the last equality of the above formula holds by the following reason. It is easy to check that for each $s\in[0,q^m)$, $\mathbf{e}_s\mathbf{e}^T_{\varphi_{u_1,v_3}(s')}+\mathbf{e}_s\mathbf{e}^T_{s'}=1$ if and only if exact one of following two conditions

$$\begin{aligned}
&\mathbf{e}_s\mathbf{e}^T_{\varphi_{u_1,v_3}(s')}=1,\quad \mathbf{e}_s\mathbf{e}^T_{s'}=0,\qquad \text{or}\\
&\mathbf{e}_s\mathbf{e}^T_{\varphi_{u_1,v_3}(s')}=0,\quad \mathbf{e}_s\mathbf{e}^T_{s'}=1,
\end{aligned}$$

holds by the fact $\mathcal{V}_{u_1,v_2}\bigcap\mathcal{V}_{u_1,v_3}=\emptyset$. By (26), if $v_1=v_3$ or $v_1=v_2$, $\mathcal{E}_{u_1,v_1}\mathbf{C}_{u_1,v_3,v_2}=\mathcal{E}_{u_1,v_2}$ can be obtained directly. Furthermore, if $v_1\neq v_2$ and $v_1\neq v_3$ we have$\mathcal{E}_{u_1,v_1}\mathbf{C}_{u_1,v_3,v_2}=\{0\}=\mathcal{E}_{u_1,v_1}\bigcap\mathcal{V}_{u_1,v_3}$.

– If $v_3=q$, we have

$$\begin{aligned}
&\mathcal{E}_{u_1,v_1}\mathbf{C}_{u_1,q,v_2}\\
=\ &\left\{\mathbf{e}_s\left(\phi_{u_1,v_2}(s')\mathbf{e}^T_{s'}\right)_{s'\in[0,q^m)}\ \middle|\ s\in\mathcal{V}_{u_1,v_1}\right\}\\
=\ &\left\{\left(\phi_{u_1,v_2}(s')\left(\mathbf{e}_s\mathbf{e}^T_{s'}\right)\right)_{s'\in[0,q^m)}\ \middle|\ s\in\mathcal{V}_{u_1,v_1}\right\}\\
=\ &\mathcal{E}_{u_1,v_1}\bigcap\mathcal{E}_{u_1,v_2}
\end{aligned}$$

So (14) always holds.

- When $u_1\neq u_2$, if $v_3<q$ for each integer $s\in\mathcal{V}_{u_1,v_1}\bigcap\mathcal{V}_{u_1,v_3}$, $\varphi_{u_1,v_2}(s)$ must be the element of $\mathcal{V}_{u_1,v_1}$ since its u_1-th entry is v_1. So $\mathcal{E}_{u_1,v_1}\mathbf{C}_{u_2,v_3,v_2}=\mathcal{E}_{u_1,v_1}\bigcap\mathcal{E}_{u_1,v_2}$ always holds by (26). If $v_3=q$, $\mathcal{E}_{u_1,v_1}\mathbf{C}_{u_2,v_3,v_2}=\mathcal{E}_{u_1,v_1}\bigcap\mathcal{E}_{u_1,v_2}$ can be directly obtained by the above formula.

Now let us verify (15) according to the value of v_3.

- When $v_3<q$, we have

$$\begin{aligned}
&\mathcal{Q}_{u_1}\mathbf{C}_{u_2,v_3,v_2}\\
&=\Big\{\mathbf{x}\left(\phi_{u_2,v_2}(s')(\mathbf{e}_{\varphi_{u_2,v_3}(s')}+\mathbf{e}_{s'})^T\right)_{s'\in[0,q^m)}\\
&\qquad\qquad\Big|\mathbf{x}\in\mathcal{Q}_{u_1}\Big\} \quad (27)\\
&=\Big\{\left(\phi_{u_2,v_2}(s')(\mathbf{x}\mathbf{e}^T_{\varphi_{u_2,v_3}(s')}+\mathbf{x}\mathbf{e}^T_{s'})\right)_{s'\in[0,q^m)}\\
&\qquad\qquad\Big|\mathbf{x}\in\mathcal{Q}_{u_1}\Big\}
\end{aligned}$$

Since each row of $\mathbf{C}_{u_2,v_3,v_2}$ has exact two entries containing 1 or has no entry containing 1, by the definition of $\mathcal{Q}_u$ in (11) we have $\mathcal{Q}_{u_1}\mathbf{C}_{u_2,v_3,v_2}=\{0\}$ if $u_1=u_2$. If $u_1\neq u_2$, we have $\mathcal{Q}_{u_1}\mathbf{C}_{u_2,v_3,v_2}=\mathcal{Q}_{u_1}$.

- When $v_3=q$, we have

$$\mathcal{Q}_{u_1}\mathbf{C}_{u_2,q,v_2}=\left\{\mathbf{x}\left(\phi_{u_2,v_2}(s')\mathbf{e}^T_{s'}\right)_{s'\in[0,q^m)}\ \middle|\ \mathbf{x}\in\mathcal{Q}_{u_1}\right\}$$

Similar to the above discussions, we have $\mathcal{Q}_{u_1}\mathbf{C}_{u_2,q,v_2}=\mathcal{Q}_{u_1}$ if $u_1\neq u_2$, and $\mathcal{Q}_{u_1}\mathbf{C}_{u_2,q,v_2}=\mathcal{E}_{u_2,v_2}$ if $u_1=u_2$.

Then the proof is completed. □

Appendix B: Proof of Theorem 2

Proof. From Construction 1, for any fixed positive integers u_1, u_2, v_1, v_2, ε_1 and ε_2, let us show that the caching matrix $\mathbf{S}_{u_1,v_1,\varepsilon_1}$, coding matrix $\mathbf{A}_{u_2,v_2,\varepsilon_2}$ and decoding matrix $\mathbf{S}'_{u_1,v_1,\varepsilon_1}$ satisfy (4) in Theorem 1. By (8), we can show our statements in the following case according to (14) and (15).

When $(u_1,v_1,\varepsilon_1)=(u_2,v_2,\varepsilon_2)$, we can show the first condition of (8), i.e., $\mathbf{S}_{u_1,v_1,\varepsilon_1}+\mathbf{S}'_{u_1,v_1,\varepsilon_1}\mathbf{A}_{u_1,v_1,\varepsilon_1}=(\mathbb{F}_2)^{q^m}$, holds. Then the first condition of (4) in Theorem I holds. Now let us consider the following subcases by (18), (19) and (20).

- If $v_1<q$, we have

$$\begin{aligned}
&\mathbf{S}'_{u_1,v_1,\varepsilon_1}\mathbf{A}_{u_1,v_1,\varepsilon_1}\\
=\ &\begin{pmatrix}\mathcal{E}_{u_1,v_1}&&\\&\ddots&\\&&\mathcal{E}_{u_1,v_1}\end{pmatrix}\begin{pmatrix}\mathbf{C}_{u_1,v_1,v_{1,1}}\\\vdots\\\mathbf{C}_{u_1,v_1,v_{1,q-z}}\end{pmatrix}_{v_{1,i}\in\mathcal{G}_{v_1,\varepsilon_1}}\\
=\ &\begin{pmatrix}\mathcal{E}_{u_1,v_1}\mathbf{C}_{u_1,v_1,v_{1,1}}\\\vdots\\\mathcal{E}_{u_1,v_1}\mathbf{C}_{u_1,v_1,v_{1,q-z}}\end{pmatrix}_{v_{1,i}\in\mathcal{G}_{v_1,\varepsilon_1},i\in[1,q-z]}\\
=\ &\sum_{v\in\mathcal{G}_{v_1,\varepsilon_1}}\mathcal{E}_{u_1,v}
\end{aligned}$$

Here the last equality in the above formula holds by the fist case of (14). So we have

$$\begin{aligned}
&\mathbf{S}_{u_1,v_1,\varepsilon}+\mathbf{S}'_{u_1,v_1,\varepsilon}\mathbf{A}_{u_1,v_1,\varepsilon}\\
=\ &\{\mathcal{E}_{u_1,v'}\mid v'\in[0,q)\setminus G_{v_1,\varepsilon_1}\}+\sum_{v\in\mathcal{G}_{v_1,\varepsilon_1}}\mathcal{E}_{u_1,v}\\
=\ &\sum_{v'\in[0,q)}\mathcal{E}_{u_1,v'}=(\mathbb{F}_2)^{q^m}
\end{aligned}$$

- If $v_1=q$, we have

$$\begin{aligned}
&\mathbf{S}'_{u_1,q,\varepsilon}\mathbf{A}_{u_1,q,\varepsilon}\\
=\ &\begin{pmatrix}\mathcal{Q}_{u_1}&&\\&\ddots&\\&&\mathcal{Q}_{u_1}\end{pmatrix}\begin{pmatrix}\mathbf{C}_{u_1,q,v_{1,1}}\\\vdots\\\mathbf{C}_{u_1,q,v_{1,q-z}}\end{pmatrix}_{v_{1,i}\in\mathcal{G}_{q-1,\varepsilon_1}}\\
=\ &\begin{pmatrix}\mathcal{Q}_{u_1}\mathbf{C}_{u_1,q,v_{1,1}}\\\vdots\\\mathcal{Q}_{u_1}\mathbf{C}_{u_1,q,v_{1,q-z}}\end{pmatrix}_{v_{1,i}\in\mathcal{G}_{q-1,\varepsilon_1}}\\
=\ &\sum_{v\in\mathcal{G}_{q-1,\varepsilon_1}}\mathcal{E}_{u_1,v}
\end{aligned}$$

Here the last equality holds by the first case of (15). So we have

$$\begin{aligned}
&\mathbf{S}_{u_1,q,\varepsilon_1} + \mathbf{S}'_{u_1,q,\varepsilon_1}\mathbf{A}_{u_1,q,\varepsilon_1}\\
=\;& \mathcal{Q}_{u_1}\bigcup\{\mathcal{E}_{u_1,v'} \mid v' \in [0,q-1)\setminus G_{q-1,\varepsilon_1}\}\\
&+ \sum_{v\in\mathcal{G}_{q-1,\varepsilon_1}} \mathcal{E}_{u_1,v}\\
=\;& \mathcal{Q}_{u_1}\bigcup\{\mathcal{E}_{u_1,v'} \mid v' \in [0,q-1)\}\\
=\;& \sum_{v'\in[0,q)} \mathcal{E}_{u_1,v'} = (\mathbb{F}_2)^{q^m}
\end{aligned}$$

When $(u_1,v_1,\varepsilon_1) \neq (u_2,v_2,\varepsilon_2)$, let us show the second condition of (8), i.e., $\mathbf{S}_{u_1,v_1,\varepsilon_1} + \mathbf{S}'_{u_1,v_1,\varepsilon_1}\mathbf{A}_{u_1,v_1,\varepsilon_1} \subseteq \mathbf{S}_{u_1,v_1,\varepsilon_1}$, holds. Then the second condition of (4) in Theorem I holds. Now let us consider the following cases by (18), (19) and (20). When $u_1 \neq u_2$, by (14) and (15), it is not difficult to obtain the following statements.

$$\mathbf{S}'_{u_1,v_1,\varepsilon_1}\mathbf{A}_{u_1,v_2,\varepsilon_2} \subseteq \mathcal{E}_{u_1,v_1}.$$

So we only need to check the case $u_1 = u_2$ in the following subcases.

- If $v_1, v_2 < q$, we have

$$\begin{aligned}
&\mathbf{S}'_{u_1,v_1,\varepsilon_1}\mathbf{A}_{u_1,v_2,\varepsilon_2}\\
=\;& \begin{pmatrix}\mathcal{E}_{u_1,v_1} & & \\ & \ddots & \\ & & \mathcal{E}_{u_1,v_1}\end{pmatrix}\begin{pmatrix}\mathbf{C}_{u_1,v_2,v_{2,1}}\\ \vdots \\ \mathbf{C}_{u_1,v_2,v_{2,q-z}}\end{pmatrix}_{v_{2,i}\in\mathcal{G}_{v_2,\varepsilon_1}}\\
=\;& \begin{pmatrix}\mathcal{E}_{u_1,v_1}\mathbf{C}_{u_1,v_2,v_{2,1}}\\ \vdots \\ \mathcal{E}_{u_1,v_1}\mathbf{C}_{u_1,v_2,v_{2,q-z}}\end{pmatrix}_{v_{2,i}\in\mathcal{G}_{v_2,\varepsilon_2}}
\end{aligned}$$

  - If $v_1 = v_2$ and $\varepsilon_1 \neq \varepsilon_2$, then by the first case of (14) we have

$$\mathbf{S}'_{u_1,v_1,\varepsilon_1}\mathbf{A}_{u_1,v_1,\varepsilon_2} = \sum_{v\in\mathcal{G}_{v_2,\varepsilon_2}} \mathcal{E}_{u_1,v}.$$

    Since $\varepsilon_1 \neq \varepsilon_2$ we have $\mathcal{G}_{v_1,\varepsilon_1}\bigcap\mathcal{G}_{v_1,\varepsilon_2} = \emptyset$. Otherwise suppose $\mathcal{G}_{v_1,\varepsilon_1}\bigcap\mathcal{G}_{v_1,\varepsilon_2} \neq \emptyset$. Then there must exist two integers, say h_1, $h_2 \in [1, q-z]$ satisfying $v_1 + h_1 + \varepsilon_1(q-z) = v_1 + h_2 + \varepsilon_2(q-z)$, i.e., $h_1 - h_2 = (\varepsilon_2 - \varepsilon_1)(q-z) \neq 0$ holds. Without loss of generality we assume that $h_1 \geq h_2$. This is impossible since $1 \leq h_1 - h_2 < q - z$. So we have $\mathcal{G}_{v_1,\varepsilon_2} \subseteq [0,q)\setminus\mathcal{G}_{v_1,\varepsilon_1}$. Then by (18),

$$\begin{aligned}
\sum_{v\in\mathcal{G}_{v_1,\varepsilon_2}} \mathcal{E}_{u_1,v} &\subseteq \{\mathcal{E}_{u_1,v'} \mid v' \in [0,q)\setminus G_{v_1,\varepsilon_1}\}\\
&\subseteq \mathbf{S}_{u_1,v_1,\varepsilon_1}
\end{aligned}$$

    always holds.
  - If $v_1 \neq v_2$, by (14) we have

$$\begin{aligned}
\mathbf{S}'_{u_1,v_1,\varepsilon_1}\mathbf{A}_{u_1,v_2,\varepsilon_2} &= \sum_{v\in\mathcal{G}_{v_2,\varepsilon_2}} \left(\mathcal{E}_{u_1,v_1}\bigcap\mathcal{E}_{u_2,v}\right)\\
&\subseteq \mathcal{E}_{u_1,v_1}.
\end{aligned}$$

- If $v_1 = v_2 = q$ and $\varepsilon_1 \neq \varepsilon_2$, by the first case of (15) we have

$$\begin{aligned}
&\mathbf{S}'_{u_1,q,\varepsilon_1}\mathbf{A}_{u_1,q,\varepsilon_2}\\
=\;& \begin{pmatrix}\mathcal{Q}_{u_1} & & \\ & \ddots & \\ & & \mathcal{Q}_{u_1}\end{pmatrix}\begin{pmatrix}\mathbf{C}_{u_1,q,v_{2,1}}\\ \vdots \\ \mathbf{C}_{u_1,q,v_{2,q-z}}\end{pmatrix}_{v_{2,i}\in\mathcal{G}_{q-1,\varepsilon_2}}\\
=\;& \begin{pmatrix}\mathcal{Q}_{u_1}\mathbf{C}_{u_1,q,v_{2,1}}\\ \vdots \\ \mathcal{Q}_{u_1}\mathbf{C}_{u_1,q,v_{2,q-z}}\end{pmatrix}_{v_{2,i}\in\mathcal{G}_{q-1,\varepsilon_2}}\\
=\;& \sum_{v\in\mathcal{G}_{q-1,\varepsilon_2}} \mathcal{E}_{u_1,v}
\end{aligned}$$

  Since $\varepsilon_1 \neq \varepsilon_2$ and $v_1 = v_2$, similar to the discussion for the case $v_1 = v_2 < q$ and $\varepsilon_1 \neq \varepsilon_2$, we also have $\mathcal{G}_{q-1,\varepsilon_2} \subseteq [0,q-1)\setminus\mathcal{G}_{q-1,\varepsilon_1}$ and then have

$$\begin{aligned}
&\sum_{v\in\mathcal{G}_{q-1,\varepsilon_2}} \mathcal{E}_{u_1,v}\\
\subseteq\;& \{\mathcal{E}_{u_1,v'} \mid v' \in [0,q-1)\setminus G_{q-1,\varepsilon_1}\} \subseteq \mathbf{S}_{u_1,q,\varepsilon_1}.
\end{aligned}$$

- If $v_1 = q$ and $v_2 < q$, by the second case of (15) we have

$$\begin{aligned}
&\mathbf{S}'_{u_1,q,\varepsilon_1}\mathbf{A}_{u_1,v_2,\varepsilon_2}\\
=\;& \begin{pmatrix}\mathcal{Q}_{u_1} & & \\ & \ddots & \\ & & \mathcal{Q}_{u_1}\end{pmatrix}\begin{pmatrix}\mathbf{C}_{u_1,v_2,v_{2,1}}\\ \vdots \\ \mathbf{C}_{u_1,v_2,v_{2,q-z}}\end{pmatrix}_{v_{2,i}\in\mathcal{G}_{v_1,\varepsilon_2}}\\
=\;& \begin{pmatrix}\mathcal{Q}_{u_1}\mathbf{C}_{u_1,v_2,v_{2,1}}\\ \vdots \\ \mathcal{Q}_{u_1}\mathbf{C}_{u_1,v_2,v_{2,q-z}}\end{pmatrix}_{v_{2,i}\in\mathcal{G}_{v_2,\varepsilon_2}}\\
=\;& \{0\} \subseteq \mathbf{S}_{u_1,q,\varepsilon_1}.
\end{aligned}$$

- If $v_1 < q$ and $v_2 = q$, by second case of (14) we have

$$\begin{aligned}
&\mathbf{S}'_{u_1,v_1,\varepsilon_1}\mathbf{A}_{u_1,q,\varepsilon_2}\\
=\;& \begin{pmatrix}\mathcal{E}_{u_1,v_1} & & \\ & \ddots & \\ & & \mathcal{E}_{u_1,v_1}\end{pmatrix}\begin{pmatrix}\mathbf{C}_{u_1,q,v_{2,1}}\\ \mathbf{C}_{u_1,q,v_{2,2}}\\ \vdots \\ \mathbf{C}_{u_1,q,v_{2,q-z}}\end{pmatrix}_{v_{2,i}\in\mathcal{G}_{q-1,\varepsilon_2}}\\
=\;& \begin{pmatrix}\mathcal{E}_{u_1,v_1}\mathbf{C}_{u_1,q-1,v_{2,1}}\\ \vdots \\ \mathcal{E}_{u_1,v_1}\mathbf{C}_{u_1,q-1,v_{2,q-z}}\end{pmatrix}_{v_{2,i}\in\mathcal{G}_{q-1,\varepsilon_2}}\\
=\;& \sum_{v\in\mathcal{G}_{q-1,\varepsilon_2}} \left(\mathcal{E}_{u_1,v_1}\bigcap\mathcal{E}_{u_1,v}\right)\\
\subseteq\;& \mathcal{E}_{u_1,v_1} \subseteq \mathbf{S}_{u_1,v_1,\varepsilon_1}.
\end{aligned}$$

□

ACKNOWLEDGMENT

The authors are very grateful to the reviewers and the Associate Editor, Prof. Krishna Narayanan, for their valuable comments that improved the quality and presentation of this paper.

REFERENCES

[1] A. Sengupta, R. Tandon, and T. Clancy, Fundamental limits of caching with secure delivery, *IEEE Trans. Inf. Foren. Sec.*, vol. 10, no. 2, pp. 355-370, 2015.

[2] M. Cheng, J. Jiang, Q. Yan, X.Tang, Coded caching schemes for flexible memory sizes, *IEEE Trans. Commun.*, vol. 67, no. 6, pp. 4166-4176, 2019.

[3] M. Cheng, J. Jiang, Q. Wang, Y. Yao, A generalized grouping scheme in coded caching, *IEEE Trans. Commun.*, vol. 67, no. 5, pp. 3422-3430, 2019.

[4] M. Cheng, J. Jiang, X. Tang, and Q. Yan, Some Variant of Known Coded Caching Schemes With Good Performance, *IEEE Trans. Commun.*, vol. 68, no.3, pp. 1370-1377, Mar. 2020.

[5] A. G. Dimakis, P. B. Godfrey, Y. Wu, M. J. Wainwright, and K. Ramchandran, Network coding for distributed storage systems, *IEEE Trans. Inform. Theory*, vol. 56, no. 9, pp. 4539-551, 2010.

[6] M. Ji, G. Caire and A. F. Molisch, Fundamental limits of caching in wireless D2D networks, *IEEE Trans. Inform. Theory*, vol. 62, no. 2, pp.849-869, 2016.

[7] M. Ji, M. Wong, A. M. Tulino, J. Llorca, On the fundamental limits of caching in combination networks, in Proc. *2015 IEEE SPAWC*, Stockholm, Sweden, July, 2015, pp. 695-699.

[8] "Kai Wan and Mingyue Ji and Pablo Piantanida and Daniela Tuninetti, Caching in Combination Networks: Novel Multicast Message Generation and Delivery by Leveraging the Network Topology, in Proc. *IEEE ICC*, Kansas City, MO, 2018, pp. 1-6.

[9] P. Krishnan, Prasad. Coded caching via line graphs of bipartite graphs, in Proc. *2018 IEEE ITW*, Guangzhou, China, Nov. 2018.

[10] N. Karamchandani, U. Niesen, M. A. Maddah-Ali, and S. Diggavi, Hierarchical coded caching, in Proc. *IEEE ISIT*, Honolulu, HI, Jun. 2014, pp. 2142-2146.

[11] M. A. Maddah-Ali and U. Niesen, Fundamental limits of caching, *IEEE Trans. Inform. Theory*, vol. 60, no. 5, pp. 2856-2867, 2014.

[12] M. A. Maddah-Ali, Urs Niesen, Decentralized coded caching attains order-optimal memory-rate tradeoff, *IEEE/ACM Trans. on Networking*, vol. 23, no. 4, pp.1029-1040, 2015.

[13] R. Pedarsani, M. A. Maddah-Ali, and U. Niesen, Online coded caching, in Proc. *IEEE ICC*, Sydney, Australia, Jun. 2014, pp. 1878-1883.

[14] C. Shangguan, Y. Zhang, G. Ge, Centralized coded caching schemes: A hypergraph theoretical approach, *IEEE Trans. Inform. Theory*, vol. 64, no. 8, pp. 5755-5766, 2018.

[15] K. Shanmugam, M. Ji, A. M. Tulino, J. Llorca, and A. G. Dimakis, Finite-length analysis of caching-aided coded multicasting, *IEEE Trans. Inform. Theory*, vol. 62, no. 10, pp. 5524–5537, 2016.

[16] K. Shanmugam, A. M. Tulino, and A. G. Dimakis, Coded caching with linear subpacketization is possible using Ruzsa-Szemredi graphs, in Proc. *IEEE ISIT*, Aachen, Germany, Jun. 2017, pp. 1237-1241.

[17] K. Shanmugam, A. G. Dimakis, J. Llorca and A. M. Tulino, A unified Ruzsa-Szemerdi framework for finite-length coded caching, 2017 51st *Asilomar Conference on Signals, Systems, and Computers*, Pacific Grove, CA, 2017, pp. 631-635.

[18] T. Tamo, Z. Wang and J. Bruck, Zigzag codes: MDS array codes with optimal rebuilding, *IEEE Trans. Inform. Theory,* vol. 59, no. 3, pp. 1597-1616, Mar. 2013.

[19] L. Tang, A. Ramamoorthy, Coded caching schemes with reduced subpacketization from linear block codes, *IEEE Trans. Inform. Theory*, vol. 64, no. 4, pp. 3099-3120, 2018.

[20] Z. Wang, T. Tamo and J. Bruck, Long MDS codes for optimal repair bandwidth, *IEEE ISIT*, Cambridge, MA, USA, July. 2012.

[21] K. Wan and D. Tuninetti and P. Piantanida, An Index Coding Approach to Caching With Uncoded Cache Placement, *IEEE Trans. Inform. Theory*, vol. 66, no. 3, pp. 1318-1332, 2020.

[22] Q. Yan, M. Cheng, X. Tang and Q. Chen, On the placement delivery array design in centralized coded caching scheme, *IEEE Trans. Inform. Theory*, vol. 63, no. 9, pp. 5821-5833, 2017.

[23] Q. Yan, X. Tang, Q. Chen, M. Cheng, Placement delivery array design through strong edge coloring of bipartite graphs, *IEEE Commun. Lett.*, vol. 22, no. 2, pp. 236-239, Feb. 2018

Minquan Cheng received the Ph.D. degree from Department of Social Systems and Management, Graduate School of Systems and Information Engineering, University of Tsukuba, Tsukuba, Ibaraki, Japan, in 2012. Then, he joined Guangxi Normal University, Guilin, Guangxi, China, where he is currently a full professor at School of Computer Science and Information Technology. His research interests include combinatorics, coding theory, cryptography and their interactions.

Jie Li (Member, IEEE) received the B.S. and M.S. degrees in mathematics from Hubei University, Wuhan, China, in 2009 and 2012, respectively, and received the Ph.D. degree from the department of communication engineering, Southwest Jiaotong University, Chengdu, China, in 2017.

From 2015 to 2016, he was a visiting Ph.D. student in the Department of Electrical Engineering and Computer Science, The University of Tennessee at Knoxville, TN, USA. From 2017 to 2019, he was a postdoctoral researcher at the Department of Mathematics, Hubei University, Wuhan, China. Since 2019, he has been a postdoctoral researcher at the Department of Mathematics and Systems Analysis, Aalto University, Finland. His research interests include coding for distributed storage, private information retrieval, and sequence design.

Dr. Li received the IEEE Jack Keil Wolf ISIT Student Paper Award in 2017.

Xiaohu Tang (M'04) received the B.S. degree in applied mathematics from the Northwest Polytechnic University, Xi'an, China, the M.S. degree in applied mathematics from the Sichuan University, Chengdu, China, and the Ph.D. degree in electronic engineering from the Southwest Jiaotong University, Chengdu, China, in 1992, 1995, and 2001 respectively.

From 2003 to 2004, he was a research associate in the Department of Electrical and Electronic Engineering, Hong Kong University of Science and Technology. From 2007 to 2008, he was a visiting professor at University of Ulm, Germany. Since 2001, he has been in the School of Information Science and Technology, Southwest Jiaotong University, where he is currently a professor. His research interests include coding theory, network security, distributed storage and information processing for big data.

Dr. Tang was the recipient of the National excellent Doctoral Dissertation award in 2003 (China), the Humboldt Research Fellowship in 2007 (Germany), and the Outstanding Young Scientist Award by NSFC in 2013 (China). He served as Associate Editors for several journals including *IEEE Transactions on Information Theory* and *IEICE Transactions on Fundamentals*, and served on a number of technical program committees of conferences.

Ruizhong Wei received the B.Sc degree in Mathematics from Suzhou University, Suzhou, China and PhD degree from University of Nebraska-Lincoln, USA, in 1998. He is currently a professor and Chair in Department of Computer Science, Lakehead University, Canada. Before joint Lakehead University, he worked as research assistant in Suzhou University, China, Mount Saint Vincent University, Canada and University of Nebraska-Lincoln, USA. He was a post-doctoral fellow in University of Waterloo, Canada from 1998 - 2000. His research interests include Combinatorial Design Theory, Combinatorics and its applications in computer networks and Coding Theory, and his research is supported by NSERC since 2001. He is a Fellow of ICA (Institute of Combinatorics and its Application).