Design of Coded Caching Schemes through Proper Orthogonal Arrays

Xianzhang Wu†, Minquan Cheng‡, Congduan Li†, Li Chen§
† School of Electronics and Communication Engineering, Sun Yat-sen University, Shenzhen 518107, China
‡ Guangxi Key Lab of Multi-source Information Mining and Security, Guangxi Normal University, Guilin 541004, China
§ School of Electronics and Information Technology, Sun Yat-sen University, Guangzhou 510006, China
Email: wuxzh7@mail2.sysu.edu.cn, chengqinshi@hotmail.com, licongd@mail.sysu.edu.cn, chenli55@mail.sysu.edu.cn

2022 IEEE International Symposium on Information Theory (ISIT) | 978-1-6654-2159-1/22/$31.00 ©2022 IEEE | DOI: 10.1109/ISIT50566.2022.9834597

***Abstract*—Coded caching is an effective technique to utilize multicasting opportunities to reduce the data transmission load in cached networks. In such a scheme, each file in the data center or library is usually divided into a number of packets to pursue a low broadcasting rate based on the designed placements at each user's cache. However, the implementation complexity of this scheme increases with the number of packets. It is crucial to design a scheme with a small subpacketization level, while maintaining a relatively low transmission rate. Recently, a combinatorial structure called placement delivery array (PDA) was proposed as an effective tool to design coded caching schemes with a low subpacketization level. This paper proposes a novel PDA construction by selecting proper orthogonal arrays (POAs). It generalizes the existing construction, making it suitable to the scenario with a more flexible memory size. Based on the proposed PDA construction, a new coded caching scheme with the coded placement is further proposed. It is shown that the proposed schemes can yield a lower subpacketization level or transmission rate over the benchmark schemes.**

***Index Terms*—Coded caching, placement delivery array, proper orthogonal array, subpacketization**

I. Introduction

The dramatic growth in the number of network users and their rising demands for video streaming services can easily cause severe network congestions during the peak hours. Coded caching system was proposed as a promising tool to reduce the data transmission load during the peak hours by utilizing the memories distributed across the network. The network model consists of a central server containing N files of equal size, which provides service to K users over an error free broadcasting channel. Each user is assumed to have a cache memory with a size of M files. Coded caching operation has two phases. First, in the placement phase, the server sends the properly designed contents to the cache of each user without knowledge of the demands. Afterwards, in the delivery phase, the server will be informed with the users' demands, and broadcast the coded packets of size R files to the users over an error free broadcasting channel. The user' demands can be satisfied with the assistance of the contents in their own caches. The quantity R is referred to as the *transmission rate* (or *rate*), i.e., the smallest number of files that must be communicated so that the demand of any user can be satisfied. A coded caching scheme is called an F-division scheme if each file can be equally divided into F packets, which is called the *subpacketization level*. If the packets are cached directly without coding in the placement phase, it is called an *uncoded placement*. Otherwise, it is called a *coded placement*.

It is known that the scheme introduced by Maddah-Ali and Niesen [1], which we refer as the MN scheme, has the optimal rate under the constraints of uncoded placement and $K \leq N$ [2]. However, its subpacketization level increases exponentially with the number of users K, which makes it impractical for large networks. Reducing the subpacketization level of the coded caching schemes has been a major problem during the past few years. It helps bridge the gap between practical implementation and theoretical studies. There exist some works on reducing the subpacketization level of the MN scheme but they trade it with an increased transmission rate [3]–[18]. E.g., Yan *et al*. [11] proposed a combinatorial structure called the placement delivery array (PDA), and showed that the MN scheme can be represented by a special PDA. Shangguan *et al*. [7] later showed that many previously existing coded caching schemes could also be represented by the appropriate PDAs. With the introduction of PDA, various coded caching schemes with a lower subpacketization level than the MN scheme were proposed in [3], [14]–[18]. Other combinatorial constructions for reducing the subpacketization level can be referred to [4], [6]–[8], [12], [13].

This paper proposes a novel construction of PDAs through the so called proper orthogonal arrays (POAs). The corresponding coded caching scheme can reduce the number of packets by a factor of q without sacrificing the transmission rate over the existing scheme of [15], where q is an integer that is greater than or equal to two. It also yields a more flexible memory size over the scheme of [16]. The proposed construction can be seen as a generalization of [16], but it requires a delicate selection of the POAs. Based on the proposed PDAs, this research finds out that some packets cached by the users have no multicasting opportunities in the delivery phase, which indicates there is no coding gain. However, if we modify the uncoded placement into the coded placement, a coded caching scheme with a smaller subpacketization level and memory ratio is further proposed.

II. Prerequisites

This section first reviews the centralized coded caching system and PDA. Then, the definition of POA will be introduced. Some key notations are introduced as follows.

Notations: Let bolded capital letters, bolded lower-case letters, and curlicue letters denote arrays, vectors, and sets, respectively. Symbol $\oplus$ represents the exclusive-or (XOR) operation. Let $\mathbb{N}^+$ denote the set of positive integers. The sets of consecutive integers are denoted as $[x:y]=\{x,x+1,\ldots,y\}$. We use $\binom{[0:m-1]}{t}$ to represent the collection of all subsets of $[0:m-1]$ with size t. Given an $l\times m$ matrix $\mathbf{F}$ and a subset $\mathcal{S}\subseteq[0:m-1]$, let $\mathbf{F}|_{\mathcal{S}}$ denote a submatrix of $\mathbf{F}$, which is obtained by taking all the columns indexed by $j\in\mathcal{S}$. Let $\mathbf{P}(i,j)$ denote the entry of array $\mathbf{P}$ with row and column indexed by i and j, respectively. Further let $(\mathbf{A}_0;\mathbf{A}_1;\ldots;\mathbf{A}_n)$ denote an array obtained by arranging arrays (or row vectors) $\mathbf{A}_0,\mathbf{A}_1,\ldots,\mathbf{A}_n$ from top to bottom. E.g., $(\mathbf{A}_0;\mathbf{A}_1)=\begin{pmatrix}\mathbf{A}_0\\ \mathbf{A}_1\end{pmatrix}$. Finally, all the vectors in examples are written as strings. E.g., (1, 0, 1, 0) is written as (1010).

A. Centralized Coded Caching System

In a centralized coded caching system, a server containing N files with equal size is connected to K users through an error free shared link, as shown in Fig.1. Each user has a cache with a size of M files, where $M<N$. The N files and K users are denoted by $\mathcal{W}=\{W_0,W_1,\ldots,W_{N-1}\}$ and $\mathcal{K}=[0:K-1]$, respectively. An F-division (K,M,N) coded caching scheme consists of two phases as follows.

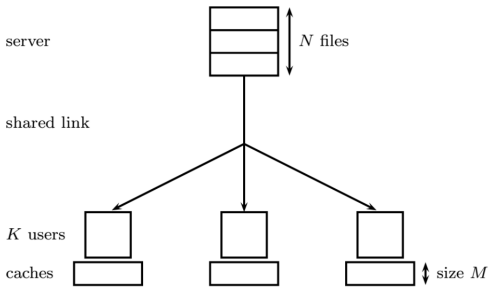

Fig. 1: Coded caching system.

• **Placement Phase**: Each file is divided into F equal packets, i.e., $W_n=\{W_{n,j}|j\in[0:F-1]\}$, $n\in[0:N-1]$. Each user can access the file set $\mathcal{W}$. Let $\mathcal{Z}_k$ denote the packet subset of $\mathcal{W}$ that is cached by user k, where $k\in\mathcal{K}$. Note that the size of $\mathcal{Z}_k$ cannot be greater than each user's cache memory size M, i.e., $|\mathcal{Z}_k|\le M$.

• **Delivery Phase**: Each user requests an arbitrary file in $\mathcal{W}$. The request vector is denoted by $\mathbf{d}=(d_0,d_1,\ldots,d_{K-1})$, i.e., user k requests file W_{d_k}, where $k\in\mathcal{K}$ and $d_k\in[0:N-1]$. Once the server receives the request vector $\mathbf{d}$, it broadcasts at most RF packets such that each user can recover its requested file together with the contents in its own cache.

B. Placement Delivery Array

Definition 1 [11]. Given $K,F,Z,S\in\mathbb{N}^+$, an $F\times K$ array $\mathbf{P}=(\mathbf{P}(i,j))$, where $i\in[0:F-1]$, $j\in[0:K-1]$, and $\mathbf{P}(i,j)\in[0:S-1]\cup\{*\}$, is called a (K,F,Z,S) PDA if the following conditions are satisfied:

C1. Symbol "$*$" appears exactly Z times in each column;

C2. Each integer of $[0:S-1]$ appears at least once in the array;

C3. For any two distinct entries $\mathbf{P}(i_1,j_1)$ and $\mathbf{P}(i_2,j_2)$, $\mathbf{P}(i_1,j_1)=\mathbf{P}(i_2,j_2)=s$ is an integer only if

(a). $i_1\neq i_2, j_1\neq j_2$, i.e., they lie in distinct rows and distinct columns;

(b). $\mathbf{P}(i_1,j_2)=\mathbf{P}(i_2,j_1)=*$, i.e., the corresponding 2×2 subarray formed by rows i_1,i_2 and columns j_1,j_2 must be in one of the following forms

$$\begin{pmatrix} s & * \\ * & s\end{pmatrix},\ \begin{pmatrix} * & s \\ s & *\end{pmatrix}.$$

E.g., the following array $\mathbf{P}$ is a (4, 2, 1, 2) PDA.

$$\mathbf{P}=\begin{pmatrix} 0 & * & 1 & * \\ * & 0 & * & 1\end{pmatrix} \tag{1}$$

Algorithm 1 Coded Caching Scheme Based on PDA [11]

1: Procedure Placement ($\mathbf{P}$, $\mathcal{W}$)
2: Split each file $W_n\in\mathcal{W}$ into F packets as $W_n=\{W_{n,j}\mid j\in[0:F-1]\}$.
3: **For** $k\in\mathcal{K}$ **do**
4: $\mathcal{Z}_k\leftarrow\{W_{n,j}\mid\mathbf{P}(j,k)=*,\forall n\in[0:N-1]\}$;
5: Procedure Delivery ($\mathbf{P},\mathcal{W},\mathbf{d}$)
6: **For** $s=0,1,\ldots,S-1$ **do**
7: Server sends $\oplus_{\mathbf{P}(j,k)=s,j\in[0:F-1],k\in[0:K-1]}W_{d_k,j}$.

Algorithm 1 has been introduced to realize the PDA based coded caching schemes in [11]. Given a (K,F,Z,S) PDA $\mathbf{P}$ with columns representing the user indices and rows representing the packet indices, if $\mathbf{P}(j,k)=*$, user k has cached the jth packet of all the files. Condition C1 of *Definition 1* implies that all the users have the same memory size and the memory ratio is $\frac{M}{N}=\frac{Z}{F}$. If $\mathbf{P}(j,k)=s$, where $s\in[0:S-1]$, the jth packet of all the files is not cached by user k. The XOR of the requested packets indicated by s will be broadcast by the server at time slot s. Condition C3 of *Definition 1* guarantees that user k can obtain its required packet, since it has cached all the other packets in the multicast message except the requested one. Finally, Condition C2 of *Definition 1* implies that the number of packets transmitted by the server is exactly S and the transmission rate is $R=\frac{S}{F}$. Furthermore, the coding gain in each time slot $s\in[0:S-1]$, denoted by g_s, equals to the occurrences of s in $\mathbf{P}$. This is because the coded packet broadcast at time slot s is useful for g_s users. Based on Algorithm 1, the following result can be obtained.

Lemma 1 [11]. Given a (K,F,Z,S) PDA, there always exists an F-division (K,M,N) coded caching scheme with a memory ratio of $\frac{M}{N}=\frac{Z}{F}$ and a transmission rate of $R=\frac{S}{F}$.

C. Orthogonal Arrays and Proper Orthogonal Arrays

Definition 2 [19]. Given any $m,q,t\in\mathbb{N}^+$ with $q\ge 2$ and $t\le m$, let $\mathbf{F}$ denote an $l\times m$ matrix defined over $[0:q-1]$. It is called an orthogonal array (OA) with a strength of t, if for each subset $\mathcal{S}\in\binom{[0:m-1]}{t}$ with size t, every t-length $(t\le m)$

row vector appears exactly $\lambda = \frac{l}{q^t}$ times in $\mathbf{F}|_{\mathcal{S}}$. It is denoted as $\mathrm{OA}_\lambda(l, m, q, t)$,

Since $l = \lambda q^t$, it can be simplified into $\mathrm{OA}_\lambda(m, q, t)$, where λ is the index of the OA. Note that if $\lambda = 1$, it can be omitted. E.g., with $m = 3, q = 2$ and $t = 2$, we can consider the following matrix

$$\mathbf{F} = (\mathbf{f}_0; \mathbf{f}_1; \mathbf{f}_2; \mathbf{f}_3) = ((110); (000); (101); (011)). \quad (2)$$

For each $\mathcal{S} \in \binom{[0:2]}{2}$, we have

$$\mathbf{F}|_{\{0,1\}} = ((11); (00); (10); (01));$$
$$\mathbf{F}|_{\{1,2\}} = ((10); (00); (01); (11));$$
$$\mathbf{F}|_{\{0,2\}} = ((10); (00); (11); (01)).$$

It can be seen that $\mathbf{F}$ in (2) satisfies *Definition 2*. It is an $\mathrm{OA}(3, 2, 2)$.

Based on the definition of OA, we also need a particular type of OA. It is called the proper OA (POA), which will enable the design of the new PDAs.

Definition 3. Given any $m, q \in \mathbb{N}^+$ with $q \geq 2$ and $m \geq 2$, an $\mathrm{OA}(m, q, m-1)$ is called a proper OA, denoted by $\mathrm{POA}(m, q, m-1)$, if the sum (mod q) of each row is a constant.

Since the POAs are crucial to our construction, we need to show the existence of POAs. In fact, it is true that the POAs always exist for any integers m and q, where $m \geq 2$ and $q \geq 2$.

Lemma 2. Let $\mathbf{F}$ denote a $q^{m-1} \times m$ matrix with the set of all rows given as

$$\mathcal{F} = \{(f_0, f_1, \ldots, f_{m-2}, x - \sum_{i=0}^{m-2} f_i) \mid f_0, f_1, \ldots, f_{m-2} \in [0 : q-1]\},$$

where $x \in [0 : q-1]$, $m \geq 2$ and $q \geq 2$, then $\mathbf{F}$ is a $\mathrm{POA}(m, q, m-1)$.

Proof: Detailed proof can be found in [20]. ■

It can be seen that the matrix of (2) is a $\mathrm{POA}(3, 2, 2)$ since the sum of each row is 0. It is worthwhile pointing out that an $\mathrm{OA}(m, q, m-1)$ is not always a $\mathrm{POA}(m, q, m-1)$. E.g., with $m = 3, q = 3$ and $t = 2$, the following matrix $\mathbf{F}$ is an $\mathrm{OA}(3, 3, 2)$, but it is not a $\mathrm{POA}(3, 3, 2)$.

$$\mathbf{F} = ((000); (011); (022); (101); (112); (120); (202); (210); (221)).$$

III. A New PDA Construction via POAs

Before introducing our construction, we first present our design intuition.

A. Design Intuition

Given a (K, F, Z, S) PDA $\mathbf{P}'$, if we replace Z_0 integers in each column of $\mathbf{P}'$ by "$*$"s, the resulting array will be a $(K, F, Z + Z_0, S_0)$ PDA, denoted by $\mathbf{P}_0$. Note that $S_0 \leq S$ always holds. This implies that the transmission rate of the scheme based on $\mathbf{P}_0$ may be smaller than the scheme based on $\mathbf{P}'$. Furthermore, we prefer to design a PDA $\mathbf{P} = (\mathbf{P}_0; \mathbf{P}_1)$ with the same memory ratio as $\mathbf{P}_0$ by adding a well designed $F_0 \times K$ array $\mathbf{P}_1$ to $\mathbf{P}_0$ without increasing S_0. This is because the new transmission rate will be smaller, as $\frac{S_0}{F+F_0} < \frac{S_0}{F}$. The following example illustrates the main idea of our construction. Given the following $(10, 5, 1, 20)$ PDA

$$\mathbf{P}' = \left(\begin{array}{ccccc|ccccc} * & 12 & 14 & 6 & 8 & * & 11 & 10 & 1 & 0 \\ 0 & * & 15 & 16 & 9 & 12 & 13 & 3 & 2 & * \\ 1 & 2 & * & 17 & 18 & 14 & 5 & 4 & * & 15 \\ 10 & 3 & 4 & * & 19 & 6 & 7 & * & 17 & 16 \\ 11 & 13 & 5 & 7 & * & 8 & * & 19 & 18 & 9 \end{array}\right),$$

if we replace its integers from 10 to 19 by "$*$"s, a new array $\mathbf{P}_0 = (\mathbf{P}_{0,0}\ \mathbf{P}_{0,1})$ can be obtained as follows.

$$\mathbf{P}_0 = \begin{array}{c} \quad\ \mathbf{P}_{0,0} \qquad\qquad\quad \mathbf{P}_{0,1} \\ \left(\begin{array}{ccccc|ccccc} * & * & * & 6 & 8 & * & * & * & 1 & 0 \\ 0 & * & * & * & 9 & * & * & 3 & 2 & * \\ 1 & 2 & * & * & * & * & 5 & 4 & * & * \\ * & 3 & 4 & * & * & 6 & 7 & * & * & * \\ * & * & 5 & 7 & * & 8 & * & * & * & 9 \end{array}\right) \end{array}.$$

Subsequently, an appropriate array $\mathbf{P}_1$ can be designed by adjusting the integers in $\mathbf{P}_0$, as shown in Fig.2. Adding the well designed array $\mathbf{P}_1$ to $\mathbf{P}_0$ from top to bottom, a new $(10, 10, 6, 10)$ PDA $\mathbf{P} = (\mathbf{P}_0; \mathbf{P}_1)$ can be obtained as follows.

$$\mathbf{P} = \left(\begin{array}{ccccc|ccccc} * & * & * & 6 & 8 & * & * & * & 1 & 0 \\ 0 & * & * & * & 9 & * & * & 3 & 2 & * \\ 1 & 2 & * & * & * & * & 5 & 4 & * & * \\ * & 3 & 4 & * & * & 6 & 7 & * & * & * \\ * & * & 5 & 7 & * & 8 & * & * & * & 9 \\ \hline * & * & * & 3 & 4 & 1 & 0 & * & * & * \\ 7 & * & * & * & 5 & 2 & * & * & * & 3 \\ 6 & 8 & * & * & * & * & * & * & 5 & 4 \\ * & 9 & 0 & * & * & * & * & 6 & 7 & * \\ * & * & 1 & 2 & * & * & 9 & 8 & * & * \end{array}\right).$$

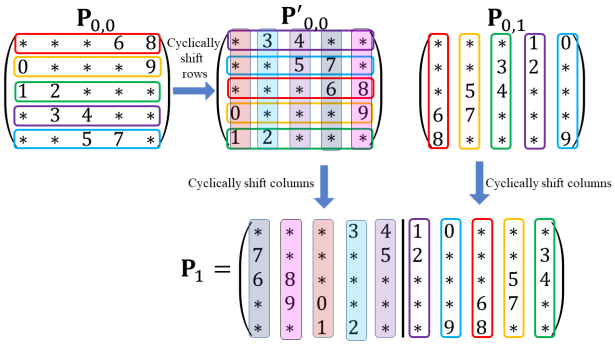

Fig. 2: The procedure of generating $\mathbf{P}_1$ from $\mathbf{P}_0$: $\mathbf{P}_0$ is divided into two 5×5 arrays, i.e., $\mathbf{P}_0 = (\mathbf{P}_{0,0}\ \mathbf{P}_{0,1})$. Then, from top to bottom, each row of $\mathbf{P}_{0,0}$ is cyclically shifted by two positions, resulting in array $\mathbf{P}'_{0,0}$. From left to right, each column of $\mathbf{P}'_{0,0}$ is further cyclically shifted by two positions, resulting in the left half of $\mathbf{P}_1$. Finally, from left to right, each column of $\mathbf{P}_{0,1}$ is cyclically shifted by two positions, resulting in the right half of $\mathbf{P}_1$.

In general, a PDA $\mathbf{P} = (\mathbf{P}_0; \mathbf{P}_1; \ldots; \mathbf{P}_L)$ constructed by the above method can be viewed as replacing the same number of integers by "$*$"s in each column of a given baseline array $\mathbf{P}'$, and adding new arrays $\mathbf{P}_1, \ldots, \mathbf{P}_L$ with the same memory ratio as $\mathbf{P}_0$ from top to bottom. In order to minimize the transmission

rate of the scheme based on $\mathbf{P}$, one needs to guarantee if an integer s in some row and column of $\mathbf{P}'$ is replaced by "$*$", all the entries of $\mathbf{P}'$ containing s are also replaced by "$*$"s. Furthermore, the newly added arrays $\mathbf{P}_1, \ldots, \mathbf{P}_L$ should be well designed such that their integers are the same as those in $\mathbf{P}_0$. This implies that the main technical challenge for the above construction is how to design a baseline array $\mathbf{P}'$ and the newly added arrays $\mathbf{P}_1, \ldots, \mathbf{P}_L$ that can satisfy such constraints.

It should be pointed out that our proposed PDA construction is different with the construction of [15]. In the construction of [15], all row indices of the newly added arrays are generated by the same OA(m, q, m) and their integers are obtained by moving the entries of a designed array in a counter clockwise manner. Furthermore, if the technique of [15] is directly applied to that of [16], the above constraints will be violated. Therefore, in order to obtain a PDA with the largest possible coding gain for each entry, some more empirical insights and technical delicacies should be utilized so that the above constraints can be satisfied.

B. New Construction

Based on the above observation, a novel framework of constructing PDA can be introduced. Let us first introduce the following notations for our construction.

• Given any $q, z, m, t \in \mathbb{N}^+$ with $z < q$ and $t < m$, let $\mathcal{E} = \{(g_0, g_1, \ldots, g_{t-1}) \mid g_i \in [0 : \lfloor \frac{q-1}{q-z} \rfloor - 1], i \in [0 : t-1]\}$, and let

$$\begin{aligned}\mathbf{F}_{\mathbf{g}_j} &= (\mathbf{f}_0^{(j)}; \mathbf{f}_1^{(j)}; \ldots; \mathbf{f}_{q^{m-1}-1}^{(j)}) \\ &= ((f_{0,0}^{(j)}, f_{0,1}^{(j)}, \ldots, f_{0,m-1}^{(j)}); (f_{1,0}^{(j)}, f_{1,1}^{(j)}, \ldots, f_{1,m-1}^{(j)}); \ldots; \\ &\quad (f_{q^{m-1}-1,0}^{(j)}, f_{q^{m-1}-1,1}^{(j)}, \ldots, f_{q^{m-1}-1,m-1}^{(j)}))\end{aligned} \tag{3}$$

denote a POA$(m, q, m-1)$ such that $\sum_{r=0}^{m-1} f_{s,r}^{(j)} = x(q-z)$, for $s \in [0 : q^{m-1} - 1]$, where $x = \sum_{i=0}^{t-1} g_i^{(j)}$, $\mathbf{g}_j = (g_0^{(j)}, g_1^{(j)}, \ldots, g_{t-1}^{(j)}) \in \mathcal{E}$, and $\mathcal{E} = \{\mathbf{g}_0, \mathbf{g}_1, \ldots, \mathbf{g}_{\lfloor \frac{q-1}{q-z} \rfloor^t - 1}\}$.

• Let $\mathscr{I} = \{\{\xi_0, \xi_1, \ldots, \xi_{t-1}\} \mid \{\xi_0, \xi_1, \ldots, \xi_{t-1}\} \in \binom{[0:m-1]}{t}, 0 \le \xi_0 < \xi_1 < \cdots < \xi_{t-1} < m\}$ and $\mathcal{C} = \{(c_0, c_1, \ldots, c_{t-1}) \mid c_i \in [0 : q-1], i \in [0 : t-1]\}$.

The new PDA construction can be proposed as follows.

Construction 1. Given any $q, z, m, t \in \mathbb{N}^+$ with $z < q$ and $t < m$, let $\mathcal{K} = \{(\mathcal{I}, \mathbf{c}) = (\{\xi_0, \xi_1, \ldots, \xi_{t-1}\}, (c_0, c_1, \ldots, c_{t-1})) \mid \mathcal{I} \in \mathscr{I}, \mathbf{c} \in \mathcal{C}\}$ and $\mathcal{F} = \bigcup_{j \in [0:\lfloor \frac{q-1}{q-z} \rfloor^t - 1]} \mathcal{F}_{\mathbf{g}_j}^{(j)}$, where $\mathcal{F}_{\mathbf{g}_j}^{(j)} = \{(\mathbf{f}_s^{(j)}, \mathbf{g}_j) = ((f_{s,0}^{(j)}, f_{s,1}^{(j)}, \ldots, f_{s,m-1}^{(j)}), (g_0^{(j)}, g_1^{(j)}, \ldots, g_{t-1}^{(j)})) \mid s \in [0 : q^{m-1} - 1]\}$ and $\mathbf{f}_s^{(j)} \in \mathbf{F}_{\mathbf{g}_j}$. An $F \times K$ array $\mathbf{P} = (\mathbf{P}_0; \ldots; \mathbf{P}_j; \ldots; \mathbf{P}_{\lfloor \frac{q-1}{q-z} \rfloor^t - 1})$ can be constructed with the entries of $\mathbf{P}_j = (\mathbf{P}_j((\mathbf{f}_s^{(j)}, \mathbf{g}_j), (\mathcal{I}, \mathbf{c})))$ defined as

$$\mathbf{P}_j((\mathbf{f}_s^{(j)}, \mathbf{g}_j), (\mathcal{I}, \mathbf{c})) = \begin{cases} (\mathbf{v}, o(\mathbf{v})), & \text{if } f_{s,\xi_h}^{(j)} \notin \{c_h, c_h - 1, \ldots, c_h - (z-1)\} \text{ for } h \in [0 : t-1]; \\ *, & \text{otherwise,} \end{cases} \tag{4}$$

where $(\mathbf{f}_s^{(j)}, \mathbf{g}_j) \in \mathcal{F}_{\mathbf{g}_j}^{(j)}$, $(\mathcal{I}, \mathbf{c}) \in \mathcal{K}$, and $\mathbf{v} = (v_0, v_1, \ldots, v_{m-1}) \in [0 : q-1]^m$ such that

$$v_i = \begin{cases} c_h - g_h^{(j)}(q-z), & \text{if } i = \xi_h, h \in [0 : t-1]; \\ f_{s,i}^{(j)}, & \text{otherwise.} \end{cases} \tag{5}$$

Note that $o(\mathbf{v})$ is the occurrence order of vector $\mathbf{v}$ in column $(\mathcal{I}, \mathbf{c})$ and the computations are performed in mod q.

The following example illustrates the above construction.

Example 1. Given $m = 2$, $q = 5$ and $t = 1$, when $z = 3$, we have $\lfloor \frac{q-1}{q-z} \rfloor = 2$ and $\mathcal{E} = \{(0), (1)\}$. Let $\mathbf{F}_{(0)}$ and $\mathbf{F}_{(1)}$ denote two POA$(2, 5, 1)$s that are defined as

$$\begin{aligned}\mathbf{F}_{(0)} &= ((00); (14); (23); (32); (41)), \\ \mathbf{F}_{(1)} &= ((02); (11); (20); (34); (43)).\end{aligned}$$

Note that the sum of each row of $\mathbf{F}_{(0)}$ is $0 \times 2 = 0$ and the sum of each row of $\mathbf{F}_{(1)}$ is $1 \times 2 = 2$. Hence, we have

$$\begin{aligned}\mathcal{F} &= (\mathbf{F}_{(0)} \times \{(0)\}) \cup (\mathbf{F}_{(1)} \times \{(1)\}); \\ \mathcal{K} &= \{(\{\xi_0\}, (c_0)) \mid \xi_0 \in [0 : 1], c_0 \in [0 : 4]\}.\end{aligned}$$

For $(\{\xi_0\}, (c_0)) = (\{0\}, (0)) \in \mathcal{K}$ and $z = 3$, we have $\{c_0, c_0 - 1, c_0 - 2)\} = \{0, 3, 4\}$. Based on (4) and (5), for any $((f_{s,0}^{(j)}, f_{s,1}^{(j)}), (g_0^{(j)})) \in \mathcal{F}$, we have $\mathbf{P}(((f_{s,0}^{(j)}, f_{s,1}^{(j)}), (g_0^{(j)})), (\{0\}, (0))) = *$, if $f_{s,0}^{(j)} \in \{0, 3, 4\}$; and $\mathbf{P}(((f_{s,0}^{(j)}, f_{s,1}^{(j)}), (g_0^{(j)})), (\{0\}, (0))) = (-2g_0^{(j)}, f_{s,1}^{(j)})$, if $f_{s,0}^{(j)} \notin \{0, 3, 4\}$. Moreover, based on (4), if $f_{s,0}^{(j)} \notin \{0, 3, 4\}$, $\mathbf{P}(((f_{s,0}^{(j)}, f_{s,1}^{(j)}), (g_0^{(j)})), (\{0\}, (0))) = (-2g_0^{(j)}, f_{s,1}^{(j)}, o((-2g_0^{(j)}, f_{s,1}^{(j)}))$. E.g., since (04) first occurs in column $(\{0\}, (0))$, and the occurrence order starts from 0, we have $\mathbf{P}(((14), (0)), (\{0\}, (0))) = (040)$. Similarly, we can obtain $\mathbf{P}(((f_{s,0}^{(j)}, f_{s,1}^{(j)}), (g_0^{(j)})), (\{\xi_0\}, (c_0)))$ for any $((f_{s,0}^{(j)}, f_{s,1}^{(j)}), (g_0^{(j)})) \in \mathcal{F}$ and $(\{\xi_0\}, (c_0)) \in \mathcal{K}$. As a result, the following PDA $\mathbf{P} = (\mathbf{P}_0; \mathbf{P}_1)$ can be obtained.

$(10, 10, 6, 10)$ PDA $\mathbf{P}$

$(\mathbf{f}, \mathbf{g}_j)/(\mathcal{I}, \mathbf{c})$	$\{0\}$					$\{1\}$				
	(0)	(1)	(2)	(3)	(4)	(0)	(1)	(2)	(3)	(4)
((00),(0))	*	*	*	(300)	(400)	*	*	*	(030)	(040)
((14),(0))	(040)	*	*	*	(440)	*	*	(120)	(130)	*
((23),(0))	(030)	(130)	*	*	*	*	(210)	(220)	*	*
((32),(0))	*	(120)	(220)	*	*	(300)	(310)	*	*	*
((41),(0))	*	*	(210)	(310)	*	(400)	*	*	*	(440)
((02),(1))	*	*	*	(120)	(220)	(030)	(040)	*	*	*
((11),(1))	(310)	*	*	*	(210)	(130)	*	*	*	(120)
((20),(1))	(300)	(400)	*	*	*	*	*	*	(210)	(220)
((34),(1))	*	(440)	(040)	*	*	*	*	(300)	(310)	*
((43),(1))	*	*	(030)	(130)	*	*	(440)	(400)	*	*

IV. Characterization and Performance

This section presents the main results of this paper, including the performance analyses of the proposed coded caching schemes.

A. Main Results

Based on *Construction 1*, this subsection presents the new coded caching schemes that are characterized in *Theorems 3* and *4*. Due to the space limit, their detailed proofs are omitted, but can be found in [20].

Theorem 3. Given any $q, z, m, t \in \mathbb{N}^+$ with $q \ge 2, z < q$ and $t < m$, there always exists an $(\binom{m}{t} q^t, \lfloor \frac{q-1}{q-z} \rfloor^t q^{m-1}, \lfloor \frac{q-1}{q-z} \rfloor^t [q^{m-1} - q^{m-t-1}(q-z)^t], q^{m-1}(q-z)^t)$ PDA which

yields a $\lfloor\frac{q-1}{q-z}\rfloor^t q^{m-1}$-division $(\binom{m}{t}q^t, M, N)$ coded caching scheme with a memory ratio of $\frac{M}{N} = 1-(\frac{q-z}{q})^t$ and a transmission rate of $R = \frac{(q-z)^t}{\lfloor\frac{q-1}{q-z}\rfloor^t}$.

The subpacketization advantage shown in *Theorem 3* depends on a delicate selection of POAs for generating the row indices. As a result, our scheme can reduce the number of packets by a factor of q without sacrificing the transmission rate over the existing scheme of [15]. It also yields a more flexible memory size over the scheme of [16].

It should be noted that there exist some useless "$*$"s (i.e., they are not contained in any subarray shown in C3-(b) of *Definition 1*) in each column of the proposed PDAs. Therefore, by deleting these useless "$*$"s and utilizing the maximum distance separable (MDS) codes in the placement phase, the following result can be obtained with a smaller subpacketization level and memory ratio than the scheme characterized *in Theorem 3*.

Theorem 4. Given any $q, m, t \in \mathbb{N}^+$ with $q \geq 2$ and $t < m$, let z_r^* denote the minimal integer in the set $\mathcal{G}_r = \{z \mid \lfloor\frac{q-1}{q-z}\rfloor = r, z \in [1:q-1]\}$, where $r \in [1:q-1]$. There exists an $(\binom{m}{t}q^t, M, N)$ coded caching scheme with a memory ratio of $\frac{M}{N} = \frac{1-(\frac{q-z_r^*}{q})^t}{1-(\frac{q-z_r^*}{q})^t+(\frac{q-z}{q})^t}$, a transmission rate of $R = \frac{(q-z)^t}{\lfloor\frac{q-1}{q-z}\rfloor^t(1-(\frac{q-z_r^*}{q})^t+(\frac{q-z}{q})^t)}$, and a subpacketization level of $F = \lfloor\frac{q-1}{q-z}\rfloor^t q^{m-1}[1+(\frac{q-z}{q})^t-(\frac{q-z_r^*}{q})^t]$.

B. Performance Analyses

We further compare our proposed schemes with the existing ones. Their features are summarized in Table I.

TABLE I
SUMMARY OF SOME KNOWN SCHEMES

Schemes and Parameters	K	$\frac{M}{N}$	R	F
MN scheme in [1], any $k, t \in \mathbb{N}^+$ with $t < k$	k	$\frac{t}{k}$	$\frac{k-t}{1+t}$	$\binom{k}{t}$
Scheme in [10], any $k, t, c \in \mathbb{N}^+$ with $t < k$	ck	$\frac{t}{k}$	$\frac{c(k-t)}{1+t}$	$\binom{k}{t}$
Scheme in [14], any $k, t \in \mathbb{N}^+$ 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
Scheme in [12], any $a, b, m, \lambda \in \mathbb{N}^+$ with $a < m, b < m$, and $\lambda < \min\{a, b\}$	$\binom{m}{a}$	$1-\frac{\binom{a}{\lambda}\binom{m-a}{b-\lambda}}{\binom{m}{b}}$	$\frac{\binom{m}{a+b-2\lambda}\binom{a+b-2\lambda}{a-\lambda}}{\binom{m}{b}}$	$\binom{m}{b}$
Scheme in [15], any $m, t, z, q \in \mathbb{N}^+$ with $t < m$, $z < q$ and $q \geq 2$	$\binom{m}{t}q^t$	$1-(\frac{q-z}{q})^t$	$\frac{(q-z)^t}{\lfloor\frac{q-1}{q-z}\rfloor^t}$	$\lfloor\frac{q-1}{q-z}\rfloor^t q^m$
Scheme in [16], any m, t and $q \in \mathbb{N}^+$ with $t < m$ and $q \geq 2$	$\binom{m}{t}q^t$	$1-(\frac{q-1}{q})^t$	$(q-1)^t$	q^{m-1}

When $z = 1$, our scheme characterized by *Theorem 3* will be the same as that of [16]. However, our scheme generalizes it into having a more flexible memory size.

In the following we compare performance of the schemes in *Theorems 3, 4* and those in [1], [10], [12], [14], [15] with $K = 300$. Let $m = 3, t = 2, q = 10$ and $z \in [1:9]$ for the schemes in *Theorems 3, 4*, and [15]; $k = 300$ and $t \in [1:299]$ for the MN scheme in [1]; $c = 10$, $k = 30$ and $t \in [1:29]$ for the scheme in [10]; $k = 25$ and $t \in \{1, 2, 22, 23\}$ for the scheme in [14]; $m = 25, a = 2, \lambda = 1$ and $b \in [1:12]$ for the scheme in [12]. Their subpacketization level F, memory ratio $\frac{M}{N}$, and transmission rate R can be characterized as in Fig. 3.

It can be seen that both the transmission rate and subpacketization level of the schemes in *Theorems 3* and *4* are closed to those of the scheme in [14], which is able to achieve the minimum subpacketization level for a fixed transmission rate. This implies that the schemes characterized by *Theorems 3* and *4* also yield a good performance. Since our proposals yield a more flexible memory size, they have a wider range of applications than the scheme of [14]. It can also be seen that with the same number of users, memory ratio and transmission rate, the scheme in *Theorem 3* has a smaller subpacketization level than that of [15]. The subpacketization level of the scheme in *Theorem 4* is even smaller than the above two, while maintaining almost the same transmission rate. When comparing with the schemes of [1], [10] and [12], our proposed schemes in *Theorems 3* and *4* have an advantage in the subpacketization level, but they are at the cost of some transmission rate.

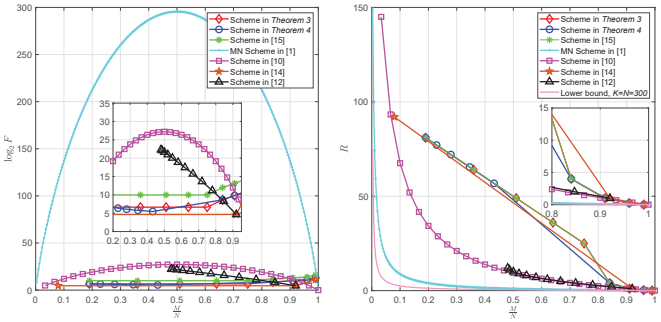

Fig. 3: Subpacketization level and transmission rate comparison between the schemes in *Theorems 3, 4* and [1], [10], [12], [14], [15], where $K = 300$.

V. Conclusion

This paper has proposed a novel construction of PDAs via POAs. Two new coded caching schemes have also been obtained, yielding a low subpacketization level and a more flexible memory size. The first PDA scheme achieves an improved subpacketization level over the existing one of [15] with the same number of users, memory ratio and transmission rate. The second PDA scheme further improves the subpacketization performance. Our analytical and numerical results have shown that the proposed schemes are able to achieve better subpacketization or transmission rate performances than the known coded caching schemes.

VI. Acknowledgement

This work is sponsored by the National Natural Science Foundation of China (NSFC) with project IDs 62071498, 61901534, U21A20474 and 62061004, the Science, Technology and Innovation Commission of Shenzhen Municipality with project ID 20190807155617099, and the Guangxi Natural Science Foundation with project ID DA035087.

References

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

[2] K. Wan, D. Tuninetti, and P. Piantanida, "On the optimality of uncoded cache placement," in Proc. *IEEE Inf. Theory Workshop (ITW)*, Cambridge, U.K., Sep. 2016, pp. 161-165.

[3] M. Zhang, M. Cheng, J. Wang, and X. Zhong, "Improving placement delivery array coded caching schemes with coded placement," *IEEE Access*, vol. 8, pp. 217456-217462, Dec. 2020.
[4] S. Agrawal, K. V. Sushena Sree, and P. Krishnan,"Coded caching based on combinatorial designs," in Proc. *IEEE Int. Symp. Inf. Theory (ISIT)*, Paris, France, Jul. 2019, pp. 1227-1231.
[5] H. H. S. Chittoor, M. Bhavana, and P. Krishnan, "Coded caching via projective geometry: A new low subpacketization scheme," in Proc. *IEEE Int. Symp. Inf. Theory (ISIT)*, Paris, France, Jul. 2019, pp. 682-686.
[6] H. H. S. Chittoor, P. Krishnan, K. V. S. Sree, and B. Mamillapalli, "Subexponential and linear subpacketization coded caching via projective geometry," *IEEE Trans. Inf. Theory*, vol. 67, no. 9, pp. 6193-6222, Sep. 2021.
[7] C. Shangguan, Y. Zhang, and G. Ge, "Centralized coded caching schemes: A hypergraph theoretical approach," *IEEE Trans. Inf. Theory*, vol. 64, no. 8, pp. 5755-5766, Aug. 2018.
[8] K. Shanmugam, A. M. Tulino, and A. G. Dimakis, "Coded caching with linear subpacketization is possible using Ruzsa-Szeméredi graphs," in Proc. *IEEE Int. Symp. Inf. Theory (ISIT)*, Aachen, Germany, Jun. 2017, pp. 1237-1241.
[9] K. Shanmugam, A. G. Dimakis, J. Llorca, and A. M. Tulino, "A unified Ruzsa-Szemerédi framework for finite-length coded caching," in Proc. The 51st ACSSC, Pacific Grove, CA, Oct. 2017, pp. 631-635.
[10] K. Shanmugam, M. Ji, A. M. Tulino, J. Llorca, and A. G. Dimakis, "Finite length analysis of caching-aided coded multicasting,"*IEEE Trans. Inf. Theory*, vol. 62, no. 10, pp. 5524-5537, Oct. 2016.
[11] Q. Yan, M. Cheng, X. Tang, and Q. Chen, "On the placement delivery array design for centralized coded caching scheme," *IEEE Trans. Inf. Theory*, vol. 63, no. 9, pp. 5821-5833, Sep. 2017.
[12] Q. Yan, X. Tang, Q. Chen, and 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.
[13] L. Tang and A. Ramamoorthy, "Coded caching schemes with reduced subpacketization from linear block codes," *IEEE Trans. Inf. Theory*, vol. 64, no. 4, pp. 3099-3120, Apr. 2018.
[14] 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.
[15] M. Cheng, J. Jiang, Q. Yan, and X. Tang, "Constructions of coded caching schemes with flexible memory size," *IEEE Trans. Commun.*, vol. 67, no. 6, pp. 4166-4176, Jun. 2019.
[16] M. Cheng, J. Wang, X. Zhong, and Q. Wang, "A framework of constructing placement delivery arrays for centralized coded caching," *IEEE Trans. Inf. Theory*, vol. 67, no. 11, pp. 7121-7131, Nov. 2021.
[17] X. Zhong, M. Cheng, and J. Jiang, "Placement delivery array based on concatenating construction," *IEEE Commun. Lett.*, vol. 24, no. 6, pp. 1216-1220, Jun. 2020.
[18] J. Michel and Q. Wang, "Placement delivery arrays from combinations of strong edge colorings," *IEEE Trans. Commun.*, vol. 68, no. 10, pp. 5953-5964, Oct. 2020.
[19] D. R. Stinson, *Combinatorial Designs: Construction and Analysis*, Springer, 2003, New York.
[20] X. Wu, M. Cheng, C. Li, and L. Chen, (Jan. 2022), "Design of placement delivery arrays for coded caching with small subpacketizations and flexible memory sizes," [Online]. Available: https://arxiv:2106.00480v2.