IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received March 6, 2020, accepted April 8, 2020, date of publication May 4, 2020, date of current version May 20, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2992091

Some New Coded Caching Schemes With Smaller Subpacketization via Some Known Results

MINQUAN CHENG, WENYU ZHANG, AND JING JIANG
Guangxi Key Laboratory of Multi-Source Information Mining and Security, Guangxi Normal University, Guilin 541004, China
Corresponding author: Jing Jiang (jjiang2008@hotmail.com)

The work of Minquan Cheng and Jing Jiang was supported in part by National Natural Science Foundation of China under Grant 11601096 and Grant 61672176, in part by Guangxi Natural Science Foundation under Grant 2016GXNSFFA380011, Grant 2016GXNSFCA380021, and Grant 2018GXNSFAA138152, in part by the Guangxi Higher Institutions Program of Introducing 100 High-Level Overseas Talents, in part by the Research Fund of Guangxi Key Laboratory of Multi-Source Information Mining and Security under Grant 16-B-01 and Grant 18-B-01, in part by the Guangxi Collaborative Innovation Center of Multi-Source Information Integration and Intelligent Processing, in part by the Guangxi Bagui Scholar Teams for Innovation and Research Project, and in part by the Guangxi Talent Highland Project of Big Data Intelligence and Application.

ABSTRACT Coded caching is widely regarded as an important tool to reduce pressure on the data transmission during the peak traffic times in heterogeneous wireless networks. In this paper, we will provide two concatenating constructions of coded caching schemes based on the schemes derived by Shangguan *et al.* in 2018 and Yan *et al.* in 2017, respectively. Then we show that our new schemes have good performance in reducing packet number. Furthermore, in some case, we present that our schemes have better performances on user number, memory ratio, packet number and transmission rate than the schemes obtained by Chittoor *et al.* in 2019 and Krishnan in 2018.

INDEX TERMS Coded caching scheme, placement delivery array, rate, packet number.

I. INTRODUCTION

The demand for high-definition video streaming services such as YouTube and Netflix is driving the rapid growth of Internet traffic. In order to mitigate the effect of this increased load on the underlying communication infrastructure, content delivery networks deploy storage memories or caches throughout the network.

Caching is an effective approach to reduce pressure on the data transmission during the peak traffic times in heterogeneous wireless networks by storing contents into memories across the network during the off-peak traffic times. Maddah-Ali and Niesen in [14] introduced coded caching scheme, which is applied in many fields due to its advantage in reducing congestion during the peak traffic times (see [8]–[10], [13], [14], [16], and references therein).

Most studies focus on a centralized (K, M, N) caching system [14], where a central server coordinates all the transmissions. The server hosting a collection of N files is connected through an error-free broadcast link to K users, each of them has a cache memory of size M files. In a coded caching scheme, there are two phases called the placement phase during off-peak times and the delivery phase during peak times. In the placement phase, all the user caches store content related to the N files. Most importantly, this is done without any prior knowledge of user requests. In the delivery phase, each user requests one of the N files from the server. Based on the requests and the stored contents of the user caches during the placement phase, the server transmits coded packets of a length of at most R files over the broadcast link to the users. Using the contents of its cache and the received coded transmissions from the server, each user could reconstruct its requested file. Formally, denote the N files by $\mathcal{W} = \{W_1, W_2, \cdots, W_N\}$ and the K users by $\mathcal{K} = \{1, 2, \cdots, K\}$. An F-division (K, M, N) coded caching scheme consists of two phases:

- **Placement Phase:** For any $1 \leq n \leq N$, the file W_n is divided into F equal packets, i.e., $W_n = \{W_{n,j} : j = 1, 2, \cdots, F\}$. For any $1 \leq k \leq K$, denote the set of packets cached by user k as $\mathcal{Z}_k$, where the size of $\mathcal{Z}_k$ is less than or equal to M.
- **Delivery Phase:** User k, $1 \leq k \leq K$, requests one file from $\mathcal{W}$, denoted by W_{d_k}. Once the server received the users' request $d = (d_1, d_2, \cdots, d_K)$, it broadcasts XOR of packets with the size at most $R_d F$ packets to users. Finally, user k, $1 \leq k \leq K$, can recover its requested file W_{d_k} with the help of received signal and the packets in its caches $\mathcal{Z}_k$.

The associate editor coordinating the review of this manuscript and approving it for publication was Jose Saldana.

TABLE 1. Summary of some known coded caching scheme.

References and parameters	Number of Users K	Cache Fraction $\frac{M}{N}$	Rate R	Subpacketization F
[4], $m, q, t, z \in Z^+$ with $t \leq m$ and $z \leq q$	$\binom{m}{t} q^t$	$1 - (\frac{q-z}{q})^t$	$\frac{(q-1)^t}{\lfloor \frac{q-1}{q-z} \rfloor^t}$	$\mathcal{O}(\lfloor \frac{q-1}{q-z} \rfloor^t q^{\frac{tK^{\frac{1}{t}}}{q}})$
	$(m+1)q$	$\frac{z}{q}$	$\frac{q-z}{\lfloor \frac{q-1}{q-z} \rfloor}$	$\lfloor \frac{q-1}{q-z} \rfloor q^{\frac{K}{q}-1}$
[11], $k, m, t \in Z^+$ with $m + t \leq k$ and q is a prime power	$\begin{bmatrix} k \\ t \end{bmatrix}_q$	$1 - \frac{\begin{bmatrix} m+t \\ t \end{bmatrix}_q}{\begin{bmatrix} k \\ t \end{bmatrix}_q}$	$\frac{\begin{bmatrix} m+t \\ t \end{bmatrix}_q}{\begin{bmatrix} k-m \\ t \end{bmatrix}_q}$	$\begin{bmatrix} k \\ m+t \end{bmatrix}_q$
[5], $k, m, t \in Z^+$ with $m + t \leq k$ and q is a prime power	$\begin{bmatrix} k-t+1 \\ 1 \end{bmatrix}_q$	$1 - \frac{q^{m+1} \begin{bmatrix} k-t \\ m+1 \end{bmatrix}_q}{\begin{bmatrix} k-t+1 \\ m+1 \end{bmatrix}_q}$	$\frac{q^{m+1} \begin{bmatrix} k-m-t \\ 1 \end{bmatrix}_q}{m+2}$	$\begin{bmatrix} k-t+1 \\ m+1 \end{bmatrix}_q \frac{\prod_{i=0}^{m} (q^{m+1} - q^i)}{(q-1)^{m+1}(m+1)!}$

TABLE 2. Summary of some latest known coded caching schemes.

References and parameters	Number of Users K	Cache Fraction $\frac{M}{N}$	Rate R	Subpacketization F
[14], $K, M, N \in Z^+$ with $M < N$ and $K\frac{M}{N} \in Z^+$	K	$\frac{M}{N}$	$\frac{K(1-\frac{M}{N})}{K\frac{M}{N}+1}$	$\binom{K}{K\frac{M}{N}}$
[26], $q, m \in Z^+$	$q(m+1)$	$1 - \frac{1}{q}$ or $\frac{1}{q}$	$\frac{1}{q-1}$ or $q - 1$	$(q-1)q^{\frac{K}{q}-1}$ or $q^{\frac{K}{q}-1}$
[27], $m, a, b, \lambda \in Z^+$ with $0 < a, b < m$ and $\lambda < \min\{a, b\}$	$\binom{m}{a}$	$\frac{\binom{a}{\lambda}\binom{m-a}{b-\lambda}}{\binom{m}{a}}$	$\frac{\binom{m}{a+b-2\lambda}\binom{a+b-2\lambda}{a-\lambda}}{\binom{m}{b}}$	$\binom{m}{b}$
[2], $k, t \in Z^+$ with $t < k$	$\binom{k}{t+1}$	$1 - \frac{t+1}{\binom{k}{t}}$	$\frac{k}{\binom{k}{t}}$	$\binom{k}{t} = K\frac{k-t}{t+1}$

Denote the maximum transmission amount among all the requests during the delivery phase by R, i.e.,

$$R = \sup_{\substack{\mathbf{d} = (d_1, \cdots, d_K) \\ d_k \in [1, N], \forall k \in [1, K]}} \{R_{\mathbf{d}}\}.$$

R is always called the rate of a coded caching scheme, which is preferable as small as possible.

A. PRIOR WORK

In this paper, we will pay our attention to the centralized coded caching schemes which many works focus on, for instances, [1], [2], [7], [12], [14], [17], [24]–[28] etc. The packet number F is one of the most important parameters in such schemes as F is finite in practice and the complexity of a coded caching scheme increases when F increases. Maddah-Ali and Niesen [14] introduced the first deterministic coded caching scheme for an F-division (K, M, N) coded caching scheme with $F = \binom{K}{KM/N}$, where KM/N is an integer. Obviously, the packet number $F = \binom{K}{KM/N}$ increases too quickly with K to be used in practice when K is large. There are some works paid attention to coded caching schemes with lower subpacketization level, for instances, [5], [11], [18], [20], [21], [26], [27] etc.

In this paper, we focus on the deterministic schemes when $K < N$. In [21], the authors pointed out that all the deterministic coded caching schemes can be recast into placement delivery array (PDA) introduced by Yan *et al.* [26] when $K \leq N$. This implies that PDA is a good tool to construct coded caching schemes. By means of PDA, Cheng *et al.* [4] generalized the constructions in [18] and [26], and obtained some schemes with more flexible memory size where F is minimum for fixed R. We list these schemes and the schemes from [5], [11] in TABLE 1, where $\begin{bmatrix} k \\ t \end{bmatrix}_q = \frac{(q^k-1)\cdots(q^{k-t+1}-1)}{(q^t-1)\cdots(q-1)}$. We also list some latest coded caching schemes in TABLE 2, which are given by Chittoor *et al.* [6]. From TABLE 2, the packet numbers of the schemes in [14] and [26] increase exponentially with the user numbers, while the rates of such schemes are optimal and asymptotically optimal, respectively. The packet number of the scheme in [27] increases exponentially, polynomially and linearly with the user numbers when the parameter a increases. The packet number of the scheme in [2] increases linearly with the user number. In [27] and [2] the authors further reduced the packet numbers by increasing rates respectively.

Clearly, we would like to construct a coded caching scheme with F as small as possible for fixed parameters K, M/N and R. In fact, we can also construct a coded caching scheme with K as large as possible for fixed parameters F, M/N and R. Concatenating construction is naturally studied when we construct a scheme with large number of users. Furthermore, given the parameters F, M/N, R, some concatenating constructions of PDAs with large number of users (i.e., large number of columns) have been studied. The first one is called grouping method [19]. Cheng *et al.* in [3] generalized the grouping method. We should point out that sometimes the new schemes obtained by generalized grouping method in [3] have better performances than the scheme directly constructed. For example, from TABLE 2 in [11], the authors listed a scheme with $K = 63$, $\frac{M}{N} = \frac{16}{21}$, $F_1 = 651$ and $R_1 = \frac{15}{7}$ when $k = 6$, $t = 1$, $q = 2$, and $m = 3$. From Lemma 3 in Section II, we have a $(K_1 = 42, 84, 64, 105)$ PDA when $(m, q, z) = (1, 21, 16)$. Applying Lemma 4 in Section II with the above PDA and $K = 63$, we have a

TABLE 3. Constructions with $M/N = \frac{z}{q}, z = 1, 2, \ldots q-1$.

Main Results	Number of Users K	Cache Fraction $\frac{M}{N}$	Rate R	Subpacketization F
Theorem 1	$\binom{m}{t} q^t (\lfloor \frac{q-1}{q-z} \rfloor)^t$	$1 - (\frac{q-z}{q})^t$	$(q-z)^t$	$\mathcal{O}(q^{\frac{tK^{\frac{1}{t}}}{q\lfloor \frac{q-1}{q-z} \rfloor}})$
Theorem 2	$q(m\lfloor \frac{q-1}{q-z} \rfloor + 1)$	$\frac{z}{q}$	$q-z$	$q^{\frac{\frac{K}{q}-1}{\lfloor \frac{q-1}{q-z} \rfloor}}$

(63, 168, 128, 315) PDA which generates a coded caching scheme with user number $K = 63$, cache size $\frac{M}{N} = \frac{16}{21}$, transmission rate $R_2 = \frac{15}{8}$, and packet number $F_2 = 168$. Clearly $R_2 = \frac{15}{8} < \frac{15}{7} = R_1$ and $F_2 = 168 < 651 = F_1$. From the above discussions, grouping method is an efficient way to reduce the packet number. However, in the grouping method, since all the users are divided into several groups with equal size and each group of users uses the same scheme, the resulting scheme has the same coded gain as the original scheme. The interested reader is referred to [3], [19] for more details. There are other results by means of concatenating constructions such as [15], [22], and so on.

B. CONTRIBUTIONS

In this paper, we will propose concatenating constructions such that the coded gain of the new scheme is lager than that of the original scheme, while the cache size increase slowly. We list the new classes of schemes in TABLE 3.

In Subsection III-C, we will show that our schemes include the scheme with $\frac{M}{N} = 1 - (\frac{q-1}{q})^t$ from [18] and the scheme with $\frac{M}{N} = \frac{1}{q}$ from [26] as special cases. Furthermore, our schemes have larger user number and smaller packet number than the schemes in [4] which are listed in Lines 2 and 3 of TABLE 1, while having larger transmission rate than those schemes in [4]. We also show that the new schemes have smaller packet number than the schemes in [5] and [11] by some examples.

The rest of this paper is organized as follows. We firstly introduce the relationship between coded caching scheme and PDA in Section II. Then two classes of the schemes in Theorems 1 and 2 are derived in Section III, and their detailed performance analyses are proposed in Section IV. Finally, we conclude the paper in Section V.

II. PLACEMENT DELIVERY ARRAY FOR CODED CACHING SCHEME

In this section, we will show the relationship between coded caching scheme and placement delivery array, which introduced by Yan *et al.* in [26].

Definition 1: ([26]) Suppose that K, F, Z and S are positive integers. $\mathbf{P} = (p_{i,j})$, $1 \le i \le F, 1 \le j \le K$, is an $F \times K$ array composed of a specific symbol " $*$ " and positive integers $1, 2, \cdots, S$, Then $\mathbf{P}$ is a (K, F, Z, S) placement delivery array (PDA for short) if

C1. the symbol " $*$ " occurs exact Z times in each column;

C2. each integer appears at least once in the array;

C3. for any two distinct entries p_{i_1,j_1} and p_{i_2,j_2}, $p_{i_1,j_1} = p_{i_2,j_2} = s$ is an integer only if

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

b. $p_{i_1,j_2} = p_{i_2,j_1} = *$, i.e., the corresponding 2×2 subarray formed by rows i_1, i_2 and columns j_1, j_2 must be of the following form

$$\begin{pmatrix} s & * \\ * & s \end{pmatrix} \text{ or } \begin{pmatrix} * & s \\ s & * \end{pmatrix}.$$

Lemma 1: ([26]) Suppose there exists a (K, F, Z, S) PDA. Then an F-division (K, M, N) coded caching scheme with caching ratio $\frac{M}{N} = \frac{Z}{F}$ and rate $R = \frac{S}{F}$ can be obtained by using Algorithm 1.

Algorithm 1 Caching Scheme Based on PDA in [26]

```
1: procedure Placement(P, $\mathcal{W}$)
2:     Split each file $W_n \in \mathcal{W}$ into $F$ packets, i.e., $W_n = \{W_{n,j} \mid j = 1, 2, \cdots, F\}$.
3:     for $k \in \mathcal{K}$ do
4:         $\mathcal{Z}_k \leftarrow \{W_{n,j} \mid p_{j,k} = *, \forall\, n = 1, 2, \cdots, N\}$
5:     end for
6: end procedure
7: procedure Delivery(P, $\mathcal{W}$, $d$)
8:     for $s = 1, 2, \cdots, S$ do
9:         Server sends $\bigoplus_{p_{j,k}=s, 1 \le j \le F, 1 \le k \le K} W_{d_k,j}$.
10:    end for
11: end procedure
```

Example 1: Consider the following (4, 4, 2, 4) PDA:

$$\mathbf{P} = \begin{pmatrix} 1 & * & * & 4 \\ 2 & 3 & * & * \\ * & 4 & 1 & * \\ * & * & 2 & 3 \end{pmatrix}.$$

Now we will show that a 4-division (4, 2, 4) coded caching scheme can be obtained by making use of Algorithm 1.

During the off-peak traffic times, i.e., the placement phase, from Line 2, for any $n \in [1, 4]$, divide each file W_n into $F = 4$ packets, i.e.,

$$W_n = \{W_{n,1}, W_{n,2}, W_{n,3}, W_{n,4}\}.$$

According to Lines 3-5, user k, $1 \le k \le 4$, caches the packets in $\mathcal{Z}_k$, where

$$\mathcal{Z}_1 = \left\{W_{n,3}, W_{n,4} \mid n \in [1, 4]\right\},$$
$$\mathcal{Z}_2 = \left\{W_{n,1}, W_{n,4} \mid n \in [1, 4]\right\},$$
$$\mathcal{Z}_3 = \left\{W_{n,1}, W_{n,2} \mid n \in [1, 4]\right\},$$
$$\mathcal{Z}_4 = \left\{W_{n,2}, W_{n,3} \mid n \in [1, 4]\right\}.$$

During the peak traffic times, i.e., the delivery phase, suppose the request vector is $\mathbf{d} = (1, 2, 3, 4)$. From Lines 8-10, we can obtain the transmitting process as following:

Intuitively, in a (K, F, Z, S) PDA $\mathbf{P}$, $p_{i,j} = *$ implies that user j has already cached the packets indexed by i of all the files in the server; $p_{i,j} = s$ for a certain integer s implies

TABLE 4. Delivery steps in Example 1.

Time Slot	Transmitted Signnal
1	$W_{1,1}\oplus W_{3,3}$
2	$W_{1,2}\oplus W_{3,4}$
3	$W_{2,2}\oplus W_{4,4}$
4	$W_{2,3}\oplus W_{4,1}$

that all the packets indexed i of all the files are not stored by user j. Hence the server should find out all the entries $p_{i_1,j_1}, p_{i_2,j_2}, \ldots, p_{i_h,j_h}$ such that $p_{i,j} = p_{i_1,j_1} = p_{i_2,j_2} = \ldots = p_{i_h,j_h} = s$, and multicast a signal $\bigoplus_{1\leq k\leq h} W_{d_{j_k},i_k}$. The property of the PDA guarantees all the users can get the requested files.

From Lemma 1, in order to obtain an F-division (K, M, N) coded caching scheme, one only need to construct an appropriate PDA. The following schemes, which are listed in TABLE 1, are the latest results on PDAs having good advantages on packet number or transmission rate.

Lemma 2: ([4]) Suppose that q, z, m and t are positive integers with $q \geq 2$, $q > z$ and $m > t$. Then there exists an $(\binom{m}{t}q^t, \lfloor\frac{q-1}{q-z}\rfloor^t q^m, \lfloor\frac{q-1}{q-z}\rfloor^t(q^m - q^{m-t}(q-z)^t), (q-z)^t q^m)$ PDA which generates a $\lfloor\frac{q-1}{q-z}\rfloor^t q^m$-division $(\binom{m}{t}q^t, M, N)$ coded caching scheme with $\frac{M}{N} = 1 - (\frac{q-z}{q})^t$ and rate $R = (q-z)^t/\lfloor\frac{q-1}{q-z}\rfloor^t$. (This is the scheme in the second row of TABLE 1)

Lemma 3: ([4]) Suppose that q, z and m are positive integers with $q \geq 2$ and $q > z$. Then there exists an $((m+1)q, \lfloor\frac{q-1}{q-z}\rfloor q^m, z\lfloor\frac{q-1}{q-z}\rfloor q^{m-1}, (q-z)q^m)$ PDA which generates a $\lfloor\frac{q-1}{q-z}\rfloor q^m$-division $((m+1)q, M, N)$ coded caching scheme with $\frac{M}{N} = \frac{z}{q}$ and rate $R = (q-z)/\lfloor\frac{q-1}{q-z}\rfloor$. (This is the scheme in the third row of TABLE 1)

In order to fit for large number of users, a generalization of grouping method was proposed by Cheng *et al.*, in [3]. That is, given a PDA which can realize a scheme with K_1 users and memory ratio M/N, one can obtain a PDA which can realize a scheme with any number of users $K \geq K_1$ and the same memory ratio M/N.

Lemma 4 (Generalizing Grouping Method [3]): Suppose there exists a (K_1, F, Z, S) PDA. Then, for any $K > K_1$, there exists a (K, h_1F, h_1Z, hS) PDA, which generates an h_1F-division (K, M, N) coded caching scheme with $\frac{M}{N} = \frac{Z}{F}$ and transmission rate $R = \frac{K}{K_1}\cdot\frac{S}{F}$, where $h_1 = \frac{K_1}{\gcd(K_1,K)}$ and $h = \frac{K}{\gcd(K_1,K)}$.

III. CONSTRUCTIONS

In this section, we first introduce the characterizations of the previously known concatenating constructions. Then we propose the main idea of our concatenating method. Finally, based on the original PDAs in [18] and [26], we obtain two new classes PDAs.

A. RESEARCH MOTIVATION

It is well known that grouping method, which was proposed by Shanmugam *et al.* in [19], is an effective method for reducing packet number. The authors in [19] obtained new PDAs from known PDAs in the following way: regard each column of the original PDA as an entirety and then combine these entries in a new array. Here we sketch the grouping method by the following useful notation. For any array $\mathbf{P} = (p_{i,j})$ with alphabet $[0, S)\bigcup\{*\}$, define $\mathbf{P}+a = (p_{i,j}+a)$ where $a + * = *$. Given a (K, F, Z, S) PDA $\mathbf{P}'$, the authors obtained an (mK, F, Z, mS) PDA

$$\mathbf{P}'' = (\mathbf{P}', \mathbf{P}' + S, \mathbf{P}' + 2S, \ldots, \mathbf{P}' + (m-1)S).$$

From the process of a coded caching scheme generated by a PDA, in each time slot s, the coded gain equals the number of occurrences of the integer s in the PDA. Clearly the number of occurrences of each integer in $\mathbf{P}''$ is the same as that of $\mathbf{P}'$. This implies that the coded gain of the schemes generated by the constructions in [19] is the same as that of the schemes generated by the original PDAs.

B. MAIN IDEA AND A RELATED EXAMPLE

From the above introduction, it is interesting to design new concatenating constructions satisfying the following conditions:

- the number of "$*$"s increases slowly;
- the number of occurrences of each integer in the resulting PDA is larger than that of the original PDA.

In this paper, we will give such a concatenating construction. Our main idea is as follows.

- Given a (K, F, Z, S) PDA $\mathbf{P}$, one can replace x integer entries of each column in $\mathbf{P}$ by "$*$"s, and derive a $(K, F, Z + x, S')$ PDA $\mathbf{P}'$.
- Construct a $(K + K', F, Z + x, S')$ PDA $(\mathbf{P}', \mathbf{P}'')$. With the number of "$*$"s in each column increasing, one has a good chance to add a well-designed $F \times K'$ array $\mathbf{P}''$ to $\mathbf{P}'$ without increasing the number S', such that each column of $\mathbf{P}''$ has the same memory ratio as the columns of $\mathbf{P}'$.

Obviously, in this case we should consider the structure features of original PDAs. So such a method is not fit for all the PDAs. In fact, the following PDAs from [18] and [26] can be used as the original PDA of our new concatenating construction.

Lemma 5: ([18]) Suppose that q, t and m are positive integers with $q \geq 2$ and $m \geq t$. Then there exists an $(\binom{m}{t}q^t, q^m, q^m - q^{m-t}(q-1)^t, (q-1)^t q^m)$ PDA with $M/N = 1 - (\frac{q-1}{q})^t$ and rate $R = (q-1)^t$.

Lemma 6: ([26]) Suppose that $q \geq 2$ and m are positive integers with $q \geq 2$. Then there exists a $(q(m+1), q^m, q^{m-1}, (q-1)q^m)$ PDA with $\frac{M}{N} = \frac{1}{q}$ and rate $R = q - 1$.

In the following subsection, we will propose a new PDA with $\frac{Z}{F} = 1 - (\frac{q-z}{q})^t$ and a new PDA with $\frac{Z}{F} = \frac{z}{q}$ for any $z \in [1, q)$ based on the PDA with $\frac{Z}{F} = 1 - (\frac{q-1}{q})^t$ from [18] and the PDA with $\frac{Z}{F} = \frac{1}{q}$ from Lemmas 5 and 6 respectively. Let us take an example to show our specific method.

Example 2: Given a (6, 9, 3, 18) PDA as follows.

$$\mathbf{P}=\left(\begin{array}{ccc|ccc}
* & (2,0,1) & (2,0,0) & * & (1,1,1) & (0,2,0)\\
* & (2,1,1) & (2,1,0) & (0,0,0) & * & (1,2,1)\\
* & (2,2,1) & (2,2,0) & (1,0,1) & (0,1,0) & *\\
\hline
(0,0,0) & * & (0,0,1) & * & (2,1,1) & (1,2,0)\\
(0,1,0) & * & (0,1,1) & (1,0,0) & * & (2,2,1)\\
(0,2,0) & * & (0,2,1) & (2,0,1) & (1,1,0) & *\\
\hline
(1,0,1) & (1,0,0) & * & * & (0,1,1) & (2,2,0)\\
(1,1,1) & (1,1,0) & * & (2,0,0) & * & (0,2,1)\\
(1,2,1) & (1,2,0) & * & (0,0,1) & (2,1,0) & *
\end{array}\right)$$

Replace the entries $(-,-,1)$ in $\mathbf{P}$ by "$*$"s, where $(-,-,1)$ means that the element in the 3th position is 1, one can derive a (6, 9, 6, 9) PDA $\mathbf{P}'$.

$$\mathbf{P}'=\left(\begin{array}{ccc|ccc}
* & * & (2,0,0) & * & * & (0,2,0)\\
* & * & (2,1,0) & (0,0,0) & * & *\\
* & * & (2,2,0) & * & (0,1,0) & *\\
\hline
(0,0,0) & * & * & * & * & (1,2,0)\\
(0,1,0) & * & * & (1,0,0) & * & *\\
(0,2,0) & * & * & * & (1,1,0) & *\\
\hline
* & (1,0,0) & * & * & * & (2,2,0)\\
* & (1,1,0) & * & (2,0,0) & * & *\\
* & (1,2,0) & * & * & (2,1,0) & *
\end{array}\right)$$

Obviously, comparing with $\mathbf{P}$, the number of "$*$"s in each column of $\mathbf{P}'$ increases. This implies that we have an opportunity to design a desired array $\mathbf{P}''$ by using the symbols in $\mathbf{P}'$. In our construction, we construct such an array based on $\mathbf{P}'$, i.e., for each $i \in [0, 2]$,

- replace the entries $(0, i, 0)$, $(1, i, 0)$ and $(2, i, 0)$ in the first three columns of $\mathbf{P}'$ with $(2, i, 0)$, $(0, i, 0)$ and $(1, i, 0)$, respectively;
- replace the entries $(i, 0, 0)$, $(i, 1, 0)$ and $(i, 2, 0)$ in the first last columns of $\mathbf{P}'$ with $(i, 2, 0)$, $(i, 0, 0)$ and $(i, 1, 0)$, respectively.

Then

$$\mathbf{P}''=\left(\begin{array}{ccc|ccc}
* & * & (1,0,0) & * & * & (0,1,0)\\
* & * & (1,1,0) & (0,2,0) & * & *\\
* & * & (1,2,0) & * & (0,0,0) & *\\
\hline
(2,0,0) & * & * & * & * & (1,1,0)\\
(2,1,0) & * & * & (1,2,0) & * & *\\
(2,2,0) & * & * & * & (1,0,0) & *\\
\hline
* & (0,0,0) & * & * & * & (2,1,0)\\
* & (0,1,0) & * & (2,2,0) & * & *\\
* & (0,2,0) & * & * & (2,0,0) & *
\end{array}\right)$$

Add $\mathbf{P}''$ to $\mathbf{P}'$, we have a (12, 9, 6, 9) PDA, i.e.,

$$(\mathbf{P}',\mathbf{P}'')=\left(\begin{array}{ccc|ccc|ccc|ccc}
* & * & (2,0,0) & * & * & (0,2,0) & * & * & (1,0,0) & * & * & (0,1,0)\\
* & * & (2,1,0) & (0,0,0) & * & * & * & * & (1,1,0) & (0,2,0) & * & *\\
* & * & (2,2,0) & * & (0,1,0) & * & * & * & (1,2,0) & * & (0,0,0) & *\\
(0,0,0) & * & * & * & * & (1,2,0) & (2,0,0) & * & * & * & * & (1,1,0)\\
(0,1,0) & * & * & (1,0,0) & * & * & (2,1,0) & * & * & (1,2,0) & * & *\\
(0,2,0) & * & * & * & (1,1,0) & * & (2,2,0) & * & * & * & (1,0,0) & *\\
* & (1,0,0) & * & * & * & (2,2,0) & * & (0,0,0) & * & * & * & (2,1,0)\\
* & (1,1,0) & * & (2,0,0) & * & * & * & (0,1,0) & * & (2,2,0) & * & *\\
* & (1,2,0) & * & * & (2,1,0) & * & * & (0,2,0) & * & * & (2,0,0) & *
\end{array}\right)$$

C. GENERALIZED CONSTRUCTION

In this subsection, we formally introduce our constructions. In order to show our constructions, the parameters K, F, Z and S of a PDA will be represented by q-ary sequences.

Construction 1: Suppose that q, z, m and t are positive integers with $0 < z < q$ and $0 < t < m$. Let

$$\begin{aligned}\mathcal{K} = \{(&\beta_0, \beta_1, \ldots, \beta_{t-1}, \gamma_0, \gamma_1, \ldots, \gamma_{t-1}, \eta_0, \ldots, \eta_{t-1}) \mid \\ &\beta_0, \ldots, \beta_{t-1} \in \mathbb{Z}_q, 0 \leq \gamma_0 < \ldots < \gamma_{t-1} < m,\\ &\eta_0, \ldots, \eta_{t-1} \in [0, \lfloor \frac{q-1}{q-z} \rfloor)\}\end{aligned} \quad (1)$$

and

$$\mathcal{F} = \{(\alpha_0, \alpha_1, \ldots, \alpha_{m-1}) \mid \alpha_0, \ldots, \alpha_{m-1} \in \mathbb{Z}_q\}. \quad (2)$$

Construct a $q^m \times \binom{m}{t} q^t \lfloor \frac{q-1}{q-z} \rfloor^t$ array $\mathbf{P}_{t,m,q,z} = (p_{a,b})$, where $a \in \mathcal{F}, b \in \mathcal{K}$ and

$$p_{a,b} = \begin{cases} (\alpha_0, \alpha_1, \ldots, \beta_i - \eta_i(q-z), & \text{if } \alpha_{\gamma_i} \notin X_{\beta_i,z},\\ \quad \ldots, \alpha_{m-1}, \alpha_{\gamma_0} - \beta_0 - 1, & \forall i \in [0, t)\\ \quad \ldots, \alpha_{\gamma_{t-1}} - \beta_{t-1} - 1)_q & \\ * & \text{otherwise} \end{cases} \quad (3)$$

where $X_{\beta_i,z} = \{\beta_i, \beta_i - 1, \ldots, \beta_i - (z-1)\}_q$ and the operations are performed modulo q.

Theorem 1: Suppose that q, z, m and t are positive integers with $0 < z < q$ and $0 < t < m$. Then the array $\mathbf{P}_{t,m,q,z}$ given in (3) is an $\binom{m}{t}\lfloor \frac{q-1}{q-z} \rfloor^t$-$(\binom{m}{t} q^t (\lfloor \frac{q-1}{q-z} \rfloor)^t, q^m, q^m - q^{m-t}(q-z)^t, (q-z)^t q^m)$ PDA with $M/N = 1 - (\frac{q-z}{q})^t$ and rate $R = (q-z)^t$.

The proof of Theorem 1 can be found in Appendix A.

Construction 2: If $t = 1$, from (1), we can obtain

$$\mathcal{K} = \{(\beta, \gamma, \eta) \mid 0 \leq \gamma < m, \beta \in \mathbb{Z}_q, \eta \in [0, \lfloor \tfrac{q-1}{q-z} \rfloor)\},$$

where q, z and m are positive integers with $z < q$. Let $\mathcal{K}_1 = \{(\beta, m) \mid \beta \in \mathbb{Z}_q\}$. Construct an array $\mathbf{H}_{m,q,z} = (\mathbf{P}, \mathbf{C})$ as follows.

- $\mathbf{P} = (p_{a,b})$, $a \in \mathcal{F}, b \in \mathcal{K}$, is the $q^m \times mq\lfloor \frac{q-1}{q-z} \rfloor$ array derived by Construction 1.
- $\mathbf{C} = (c_{a,b})$, $a \in \mathcal{F}, b \in \mathcal{K}_1$ is a $q^m \times q$ array define as follows.

$$c_{a,b} = \begin{cases} * & \text{if } \sum_{k=0}^{m-1} \alpha_k \in Y_{\beta,z}\\ (\alpha_0, \cdots, \alpha_{m-1}, & \text{otherwise}\\ \beta - \sum_{k=0}^{m-1} \alpha_k - 1)_q & \end{cases} \quad (4)$$

where $Y_{\beta,z} = \{\beta, \beta+1, \ldots, \beta+(z-1)\}_q$ and the operations are performed modulo q.

Theorem 2: Suppose that q, z and m with $q \geq 2$ and $q > z$. Then the array $\mathbf{H}_{m,q,z}$ generated by Construction 2 is an $(m\lfloor \frac{q-1}{q-z} \rfloor + 1)$-$(q(m\lfloor \frac{q-1}{q-z} \rfloor + 1), q^m, zq^{m-1}, (q-z)q^m)$ PDA with $\frac{M}{N} = \frac{z}{q}$ and rate $R = q - z$.

The proof of Theorem 2 can be found in Appendix B.

Example 3: When $m = 1$ and $q = 5$, we can obtain the following $\mathbf{H}_{1,5,z}$ PDAs by Construction 2 for $z = 1, 2, 3, 4$ respectively.

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

$$\mathbf{H}_{1,5,2} = \left(\begin{array}{ccccc|ccccc} * & * & 12 & 8 & 4 & * & 0 & 5 & 10 & * \\ 0 & * & * & 13 & 9 & * & * & 1 & 6 & 11 \\ 5 & 1 & * & * & 14 & 12 & * & * & 2 & 7 \\ 10 & 6 & 2 & * & * & 8 & 13 & * & * & 3 \\ * & 11 & 7 & 3 & * & 4 & 9 & 14 & * & * \end{array}\right)$$

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

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

$$\mathbf{H}_{1,5,4} = \left(\begin{array}{ccccc|ccccc|} * & * & * & * & 4 & * & * & * & * & 3 \\ 0 & * & * & * & * & 4 & * & * & * & * \\ * & 1 & * & * & * & * & 0 & * & * & * \\ * & * & 2 & * & * & * & * & 1 & * & * \\ * & * & * & 3 & * & * & * & * & 2 & * \end{array}\right.$$

$$\left.\begin{array}{ccccc|ccccc|ccccc} * & * & * & * & 2 & * & * & * & * & 1 & * & 0 & * & * & * \\ 3 & * & * & * & * & 2 & * & * & * & * & * & * & 1 & * & * \\ * & 4 & * & * & * & * & 3 & * & * & * & * & * & * & 2 & * \\ * & * & 0 & * & * & * & * & 4 & * & * & * & * & * & * & 3 \\ * & * & * & 1 & * & * & * & * & 0 & * & 4 & * & * & * & * \end{array}\right)$$

Remark 1: It is worth noting that when $z = 1$, our new PDAs in Theorems 1 and 2 are exactly the PDAs in Lemmas 5 and 6, respectively. This implies that these two families of PDAs from [18] and [26] can be obtained by our method.

IV. PERFORMANCE ANALYSES

In this section, we compare our new constructions with schemes obtained by generalizing grouping method from Lemma 4 and the schemes from [5], [11] which are the most latest results.

TABLE 5. The values of F_{Th1} and F_{G_1} when $q = 5$ and $z = 3$.

m	t	F_{Th1}	F_{G_1}
5	3	3.13×10^3	2.50×10^4
6	4	1.56×10^3	2.50×10^5
7	5	7.81×10^4	2.50×10^6
8	6	3.91×10^5	2.50×10^7
9	7	1.95×10^6	2.50×10^8
10	8	9.77×10^6	2.50×10^9

A. COMPARISON OF THE SCHEMES FROM THEOREM 1 AND GENERALIZING GROUPING METHOD

Suppose that m, t, q and z are positive integers satisfying $m > t$ and $q > z$. We compare performances of the scheme from Theorem 1 and the scheme generated by generalizing grouping method based the original scheme from Lemma 2.

From Theorem 1, we have an $({m \choose t} q^t (\lfloor \frac{q-1}{q-z} \rfloor)^t$, q^m, $q^m - q^{m-t}(q-z)^t$, $(q-z)^t q^m)$ PDA, which generates a (K, M, N) scheme, say Scheme 1, with

$$K = (\lfloor \frac{q-1}{q-z} \rfloor)^t {m \choose t} q^t, \quad \frac{M}{N} = 1 - (\frac{q-z}{q})^t,$$
$$F_{Th1} = q^m, \quad R_{Th1} = (q-z)^t. \tag{5}$$

From Lemma 2, we have a (K_1, F, Z, S) PDA where

$$K_1 = {m \choose t} q^t, \quad F = \lfloor \frac{q-1}{q-z} \rfloor^t q^m, \; S = (q-z)^t q^m,$$
$$Z = \lfloor \frac{q-1}{q-z} \rfloor^t (q^m - q^{m-t}(q-z)^t), \tag{6}$$

which generates a $\lfloor \frac{q-1}{q-z} \rfloor^t q^m$-division $({m \choose t} q^t, M, N)$ coded caching scheme with $\frac{M}{N} = 1 - (\frac{q-z}{q})^t$ and rate $R = (q-z)^t / \lfloor \frac{q-1}{q-z} \rfloor^t$. For any positive integer $K > K_1$, from Lemma 4, we have a $(K, h_1 F, h_1 Z, hS)$ PDA, which generates an $h_1 F$-division (K, M, N) coded caching scheme, say Scheme 2, with $\frac{M}{N} = \frac{Z}{F}$ and rate $R = \frac{K}{K_1} \cdot \frac{S}{F}$, where $h_1 = \frac{K_1}{\gcd(K_1, K)}$ and $h = \frac{K}{\gcd(K_1, K)}$. Let $K = \lfloor \frac{q-1}{q-z} \rfloor^t {m \choose t} q^t$. Then we have $h_1 = 1$ and $h = \lfloor \frac{q-1}{q-z} \rfloor^t$. By (6), the parameters of Scheme 2 are:

$$K = \lfloor \frac{q-1}{q-z} \rfloor^t {m \choose t} q^t, \quad \frac{M}{N} = 1 - (\frac{q-z}{q})^t,$$
$$F_{G_1} = \lfloor \frac{q-1}{q-z} \rfloor^t q^m, \quad R_{G_1} = (q-z)^t. \tag{7}$$

By (5) and (7) we have

$$\frac{F_{Th1}}{F_{G_1}} = \frac{q^m}{\lfloor \frac{q-1}{q-z} \rfloor^t q^m} = \frac{1}{\lfloor \frac{q-1}{q-z} \rfloor^t},$$
$$\frac{R_{Th1}}{R_{G_1}} = \frac{(q-z)^t}{(q-z)^t} = 1.$$

From the above formulas, we can see $R_{G_1} = R_{Th1}$ and $F_{G_1} \geq F_{Th1}$. This implies that if $\lfloor \frac{q-1}{q-z} \rfloor > 1$, Scheme 1 has the same user number, memory ratio and transmission rate while has smaller packet number than Scheme 2. We list some examples with $(q, z) = (5, 3)$ in the following table.

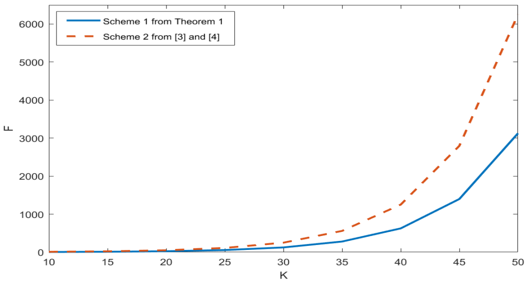

FIGURE 1. The packet numbers of Scheme 1 and 2 when $t = 1$, $q = 5$ and $z = 3$.

When $t = 1$, $q = 5$ and $z = 3$, according to (5) and (7), we have $K = 10m$, $\frac{M}{N} = \frac{3}{5}$, $R_{Th1} = R_{G_1} = 2$, $F_{Th1} = 5^m$, $F_{G_1} = 2 \cdot 5^m$. We list F_{Th1} and F_{G_1} in FIGURE 1.

Similarly we can also compare performances of the scheme from Theorem 1 and the scheme generated by generalizing grouping method based the original scheme from Lemma 3.

B. COMPARISON OF THE SCHEMES FROM THEOREM 2 AND GENERALIZING GROUPING METHOD

Suppose that m, q and z are positive integers satisfying $q > z$. We now compare performances of the scheme from Theorem 2 and the scheme generated by generalizing grouping method based the original scheme from Lemma 3.

From Theorem 2, we have a $(q(m\lfloor\frac{q-1}{q-z}\rfloor + 1), q^m, zq^{m-1}, (q-z)q^m)$ PDA, which generates a (K, M, N) scheme, say Scheme 3, with

$$K = q(m\lfloor\frac{q-1}{q-z}\rfloor + 1), \quad \frac{M}{N} = \frac{z}{q},$$
$$F_{Th2} = q^m, \quad R_{Th2} = q - z. \tag{8}$$

From Lemma 3, we have a (K_1, F, Z, S) PDA where

$$K_1 = (m+1)q, \quad F = \lfloor\frac{q-1}{q-z}\rfloor q^m,$$
$$Z = z\lfloor\frac{q-1}{q-z}\rfloor q^{m-1}, \quad S = (q-z)q^m, \tag{9}$$

which generates a $\lfloor\frac{q-1}{q-z}\rfloor q^m$-division $((m+1)q, M, N)$ coded caching scheme with $\frac{M}{N} = \frac{z}{q}$ and rate $R = (q - z)/\lfloor\frac{q-1}{q-z}\rfloor$. For any positive integer $K > K_1$, from Lemma 4, we have a (K, h_1F, h_1Z, hS) PDA, which generates an h_1F-division (K, M, N) coded caching scheme, say Scheme 4, with $\frac{M}{N} = \frac{Z}{F}$ and $R = \frac{K}{K_1} \cdot \frac{S}{F}$, where $h_1 = \frac{K_1}{\gcd(K_1, K)}$ and $h = \frac{K}{\gcd(K_1, K)}$. Let $K = q(\lfloor\frac{q-1}{q-z}\rfloor m + 1)$. We have $h_1 = \frac{m+1}{\gcd(m\lfloor\frac{q-1}{q-z}\rfloor+1, m+1)}$ and $h = \frac{m\lfloor\frac{q-1}{q-z}\rfloor+1}{\gcd(m\lfloor\frac{q-1}{q-z}\rfloor+1, m+1)}$. By (9), the parameters of Scheme 4 are:

$$K = q(\lfloor\frac{q-1}{q-z}\rfloor m + 1), \quad \frac{M}{N} = \frac{z}{q},$$
$$F_{G_2} = \frac{m+1}{\gcd(m\lfloor\frac{q-1}{q-z}\rfloor + 1, m+1)}\lfloor\frac{q-1}{q-z}\rfloor q^m,$$

TABLE 6. The values of F_{Th2}, F_{G_2}, R_{Th2} and R_{G_2} when $m = 20$.

q	z	F_{Th2}	F_{G_2}	R_{Th2}	R_{G_2}
5	3	9.53×10^{13}	4.00×10^{15}	2.00	1.95
6	4	3.66×10^{15}	3.83×10^{17}	1.00	0.96
7	5	7.98×10^{16}	5.03×10^{18}	2.00	1.94
8	6	1.15×10^{18}	7.26×10^{19}	2.00	1.94
9	6	1.22×10^{19}	5.00×10^{20}	3.00	2.93
10	9	1.00×10^{20}	1.89×10^{22}	1.00	0.96

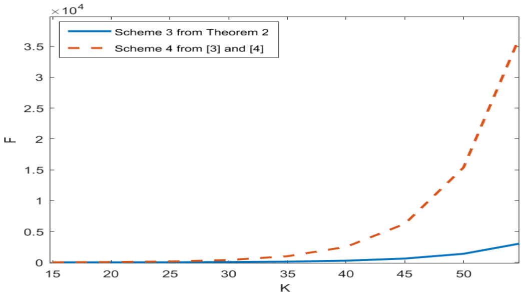

FIGURE 2. The packet numbers of Scheme 3 and 4 when $t = 1$, $q = 5$ and $z = 3$.

$$R_{G_2} = \frac{q-z}{\lfloor\frac{q-1}{q-z}\rfloor}\frac{m\lfloor\frac{q-1}{q-z}\rfloor + 1}{m+1}. \tag{10}$$

By (8) and (10), we have

$$\frac{R_{G_2}}{R_{Th2}} = 1 - \frac{\lfloor\frac{q-1}{q-z}\rfloor - 1}{\lfloor\frac{q-1}{q-z}\rfloor}\frac{1}{m+1},$$
$$\frac{F_{G_2}}{F_{Th2}} = \frac{m+1}{\gcd(m\lfloor\frac{q-1}{q-z}\rfloor + 1, m+1)}\lfloor\frac{q-1}{q-z}\rfloor. \tag{11}$$

When $gcd(m\lfloor\frac{q-1}{q-z}\rfloor + 1, m+1) = 1$, for instance, $m+1$ is a prime power,

$$\frac{F_{G_2}}{F_{Th2}} = (m+1)\lfloor\frac{q-1}{q-z}\rfloor. \tag{12}$$

Obviously, when m is large, $\frac{R_{G_2}}{R_{Th2}}$ approximates to 1, and F_{G_2} is much larger than F_{Th2}. In such a case, Scheme 3 has the same user number, memory ratio and almost the same transmission rate while has much smaller packet number than Scheme 4. We list some examples with $m = 20$ in the following table.

When $t = 1$, $q = 5$ and $z = 3$, according to (8) and (10), we have $K = 10m + 5$, $\frac{M}{N} = \frac{3}{5}$, $R_{Th2} = 2$, $R_{G_2} = \frac{2m+1}{m+1}$, $F_{Th2} = 5^m$ and $F_{G_2} = 2 \cdot 5^m$. We list F_{Th2} and F_{G_2} in FIGURE 2, and R_{Th2} and R_{G_2} in FIGURE 3.

C. COMPARISON OF THE SCHEMES FROM THEOREMS 1,2 AND [5], [11]

In this subsection, we compare performances of the schemes in Theorems 1, 2 and [5], [11]. In fact, we can not propose a theoretic analysis as the expressions of parameters in the schemes from [5] and [11] are too complex. So we only make comparisons by some examples listed in TABLE 7.

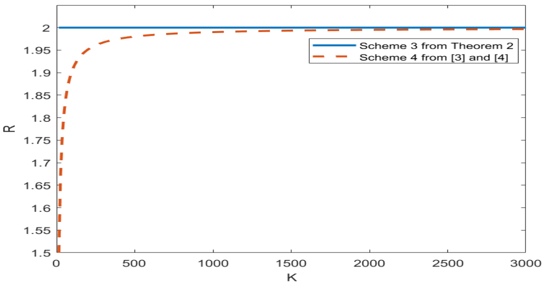

FIGURE 3. The rates of Scheme 3 and 4 when $t = 1$, $q = 5$ and $z = 3$.

TABLE 7. The schemes in Theorem 1,2 and [5], [11].

Schemes and parameters	K	$\frac{M}{N}$	R	F
$(k, m, t, q) = (7, 3, 3, 3)$ in [5]	341	0.249	51.2	5.19×10^6
$(t, m, q, z) = (2, 4, 8, 1)$ in Theorem 1	384	0.234	49.0	4.10×10^3
$(t, m, q, z) = (2, 5, 8, 1)$ in Theorem 1	640	0.234	49.0	4.10×10^3
$(k, m, t, q) = (6, 3, 2, 2)$ in [5]	31	0.490	3.2	2.60×10^4
$(m, q, z) = (6, 5, 2)$ in Theorem 2	35	0.400	3.0	1.56×10^4
$(m, q, z) = (7, 4, 1)$ in Theorem 2	32	0.250	3.0	1.64×10^4
$(k, m, t, q) = (8, 4, 1, 2)$ in [11]	127	0.843	4.4	2.67×10^3
$(t, m, q, z) = (1, 2, 16, 13)$ in Theorem 1	160	0.813	3.0	2.56×10^2
$(t, m, q, z) = (2, 7, 3, 1)$ in Theorem 1	189	0.556	4.0	2.19×10^3
$(k, m, t, q) = (6, 3, 1, 2)$ in [11]	63	0.762	2.1	6.51×10^2
$(m, q, z) = (3, 7, 5)$ in Theorem 2	70	0.714	2.0	3.43×10^2
$(m, q, z) = (3, 8, 6)$ in Theorem 2	80	0.750	2.0	5.12×10^2

- From Lines 2-4 (or Lines 5-7) in TABLE 7, the scheme in Theorem 1 (or Theorem 2) with parameters $(t, m, q, z) \in \{(2, 4, 8, 1), (2, 5, 8, 1)\}$ (or $(m, q, z) \in \{(6, 5, 2), (7, 4, 1)\}$) has larger user number than the scheme with $(t, m, q, z) = (7, 3, 3, 3)$ (or $(t, m, q, z) = (6, 3, 2, 2)$) in [5] while has smaller memory ratio, packet number and transmission rate than the scheme in [5]. That is, such schemes in Theorem 1 (or Theorem 2 has better performances on user number, memory ratio, packet number and transmission rate than the scheme in [5]. It is worth noting that the results in [5] is the latest works on low subpacketization scheme.
- From Lines 8-10 (or Lines 11-13) in TABLE 7, the scheme in Theorem 1 (or Theorem 2) with parameters $(t, m, q, z) \in \{(1, 2, 16, 13), (2, 7, 3, 1)\}$ (or $(m, q, z) \in \{(3, 7, 5), (3, 8, 6)\}$) has better performances on user number, memory ratio, packet number and transmission rate than the scheme with $(k, m, t, q) = (8, 4, 1, 2)$ (or $(k, m, t, q) = (6, 3, 1, 2)$) from [11].

V. CONCLUSION

In this paper, two concatenating constructions of coded caching schemes are obtained. Then we showed that our schemes can effectively reduce packet number. Furthermore, in some case, our schemes have better performances on user number, memory ratio, packet number and transmission rate than the schemes in [5] and [11].

The concatenating constructions in this paper are based on the PDAs in [18] and [26]. However, such constructions are not suitable for all the known PDAs. So it is interesting to give special concatenating constructions for some known PDAs, such that the coded gain of the new scheme is lager than that of the original scheme, while the cache size increase slowly.

APPENDIX A: PROOF OF THEOREM 1

Proof: We can directly check that there are exact $q^{m-t}(q-z)^t$ integer entries in each column, i.e., $Z = q^m - q^{m-t}(q-z)^t$. So C1 holds. By (3), the integer set of $\mathbf{P}_{t,m,q,z}$ in Construction 1 is

$$\begin{aligned}\mathcal{S} = \{&(s_0, s_1, \ldots, s_{m-1}, s_m, \ldots, s_{m+t-1}) \mid \\ &s_0, \ldots, s_{m-1} \in \mathbb{Z}_q, s_m, \ldots, s_{m+t-t} \in [0, q-z)\}\end{aligned}$$

So $S = |\mathcal{S}| = q^m(q-z)^t$. For an entry $p_{a,b}$ in $\mathbf{P}_{t,m,q,z}$, there exists $s = (s_0, s_1, \ldots, s_{m+t-1}) \in \mathcal{S}$ such that $p_{a,b} = s$ if and only if

$$\begin{aligned}a &= (\alpha_0, \ldots, \alpha_{\gamma_i}, \cdots, \alpha_{m-1}) \\ &= (s_0, \ldots, s_{\gamma_i} + s_{m+i} + \eta_i(q-z) + 1, \cdots, s_{m-1})_q \\ &\qquad\qquad \text{and} \\ b &= (\beta_0, \beta_1, \ldots, \beta_{t-1}, \gamma_0, \gamma_1, \ldots, \gamma_{t-1}, \eta_0, \ldots, \eta_{t-1}) \\ &= (s_{\gamma_0} + \eta_0(q-z), s_{\gamma_1} + \eta_1(q-z), \ldots, \\ &\quad s_{\gamma_{t-1}} + \eta_{t-1}(q-z), \gamma_0, \gamma_1, \ldots, \gamma_{t-1}, \\ &\quad \eta_0, \ldots, \eta_{t-1}). \qquad (13)\end{aligned}$$

for any t subset $\mathcal{T} = \{\gamma_0, \gamma_1, \ldots, \gamma_{t-1}\} \subseteq [0, m)$ and t subset $\Omega = \{\eta_0, \eta_1, \ldots, \eta_{t-1}\} \subseteq [0, \lfloor \frac{q-1}{q-z} \rfloor)$. Since there exists exact a unique pair (a, b), such that $p_{a,b} = s$ for any fixed $\mathcal{T}$ and Ω, then s occurs at most once in each column of $\mathbf{P}_{t,m,q,z}$. Furthermore, there exist exact $\binom{m}{t}(q-z)^t$ pairs $(\mathcal{T}, \Omega)$, then integer s occurs $\binom{m}{t}(q-z)^t$ times in $\mathbf{P}_{t,m,q,z}$, i.e, C2 holds.

From the above discussion, we only verify whether or not integer s occurs at most once in each row and C3-b) holds. Assume there exists $(a', b') \neq (a, b)$ such that $p_{a',b'} = s$. Then there must exist $(\mathcal{T}', \Omega')$ where $\mathcal{T}' = \{\gamma'_0, \ldots, \gamma'_{t-1}\}$ and $\Omega' = \{\eta'_0, \ldots, \eta'_{t-1}\}$, satisfying

$$\begin{aligned}a' &= (\alpha'_0, \ldots, \alpha'_{\gamma'_i}, \cdots, \alpha'_{m-1}) \\ &= (s_0, \ldots, s_{\gamma'_i} + s_{m+i} + \eta'_i(q-z) + 1, \cdots, s_{m-1})_q \\ &\qquad \text{and} \\ b' &= (\beta'_0, \beta'_1, \ldots, \beta'_{t-1}, \gamma'_0, \gamma'_1, \ldots, \gamma'_{t-1}, \eta'_0, \ldots, \eta'_{t-1}) \\ &= (s_{\gamma'_0} + \eta'_0(q-z), s_{\gamma'_1} + \eta'_1(q-z), \ldots, \\ &\quad s_{\gamma'_{t-1}} + \eta'_{t-1}(q-z), \gamma'_0, \gamma'_1, \ldots, \gamma'_{t-1}, \\ &\quad \eta'_0, \ldots, \eta'_{t-1}). \qquad (14)\end{aligned}$$

We only need to consider the following cases, where all the operations are performed modulo q.

- If $\{\gamma_0, \ldots, \gamma_{t-1}\} \neq \{\gamma'_0, \ldots, \gamma'_{t-1}\}$, then we can found out $i, i' \in [0, t)$ such that $\gamma_i \notin \{\gamma'_0, \ldots, \gamma'_{t-1}\}$ and $\gamma'_{i'} \notin \{\gamma_0, \ldots, \gamma_{t-1}\}$. Next, we will prove that C3-a) and C3-b) hold respectively.
  - If $a = a'$, which implies s appears in the row a at least twice, then $\alpha_{\gamma_i} = \alpha'_{\gamma_i}$. According to (13) and (14), we can obtain

$$\alpha_{\gamma_i} = s_{\gamma_i} + s_{m+i} + \eta_i(q-z) + 1, \quad \alpha'_{\gamma_i} = s_{\gamma_i}.$$

This implies that

$$s_{m+i} + \eta_i(q-z) + 1 = 0. \tag{15}$$

Since $s_{m+i} \in [0, q-z)$ and $\eta_i \in [0, \lfloor \frac{q-1}{q-z} \rfloor)$, then

$$\begin{aligned} 1 &\le s_{m+i} + \eta_i(q-z) + 1 \\ &< q - z + (\frac{q-1}{q-z} - 1)(q-z) + 1 = q, \end{aligned}$$

i.e., $s_{m+i}+\eta_i(q-z)+1 < q$, a contradiction to (15). Hence s occurs at most once in each row. So C3-a) holds.

– If $a \neq a'$, from (13) and (14) we have

$$\alpha_{\gamma'_{i'}} = s_{\gamma'_{i'}}, \quad \beta'_{i'} = s_{\gamma'_{i'}} + \eta'_{i'}(q-z).$$

Assume $p_{a,b'} \neq *$. Then $\alpha_{\gamma'_{i'}} \in \{\beta'_{i'}+1, \beta'_{i'}+2, \ldots, \beta'_{i'}+(q-z)\}$. That is, there is an integer $j \in [1, q-z)$ such that $\alpha_{\gamma'_{i'}} = \beta'_{i'} + j$, i.e.,

$$s_{\gamma'_{i'}} = \alpha_{\gamma'_{i'}} = \beta'_{i'} + j = s_{\gamma'_{i'}} + \eta'_{i'}(q-z) + j.$$

This implies that

$$\eta'_{i'}(q-z) + j = 0. \tag{16}$$

Since $\eta'_{i'} \in [0, \lfloor \frac{q-1}{q-z} \rfloor)$, then

$$\begin{aligned} 1 &\le \eta'_{i'}(q-z) + j \\ &< (\frac{q-1}{q-z} - 1)(q-z) + (q-z) = q-1, \end{aligned}$$

i.e., $\eta'_{i'}(q-z)+j < q-1$, a contradiction to (16). Hence $p_{a,b'} = *$. Similarly we can prove $p_{a',b} = *$. So C3-b) holds.

- If $\{\gamma_0, \ldots, \gamma_{t-1}\} = \{\gamma'_0, \ldots, \gamma'_{t-1}\}$, we have $(\eta_0, \ldots, \eta_{t-1}) \neq (\eta'_0, \ldots, \eta'_{t-1})$ since s occurs at most once in each column. So there is an integer i satisfying $\eta_i \neq \eta'_i$. Then we have

$$\begin{aligned} &\beta_i = s_{\gamma_i} + \eta_i(q-z), \quad \alpha_{\gamma_i} = s_{m+i} + b_i + 1, \\ &\beta'_i = s_{\gamma_i} + \eta'_i(q-z). \end{aligned} \tag{17}$$

– $a = a'$ does not happen. From the above investigation, i.e., there exists exact a unique pair (a, b), such that $p_{a,b} = s$ for any fixed $\mathcal{T}$ and Ω, s does not occur in the same row. Then C3-a) always holds.

– Suppose that $a \neq a'$. If $p_{a,b'} \neq *$, then $\alpha_{\gamma_i} \in \{\beta'_i + 1, \beta'_i+2, \ldots, \beta'_i+(q-z)\}$. Hence there is an integer $j \in [1, q-z)$ satisfying $\alpha_{\gamma_i} = \beta'_i + j$. Together with (17), we have

$$\begin{aligned} &s_{m+i} + s_{\gamma_i} + \eta_i(q-z) + 1 \\ &\quad = s_{\gamma_i} + \eta'_i(q-z) + j. \end{aligned} \tag{18}$$

If $\eta_i < \eta'_i$, we have

$$s_{m+i} + 1 = (\eta'_i - \eta_i)(q-z) + j. \tag{19}$$

Since $0 \le \eta_i, \eta'_i < \lfloor \frac{q-1}{q-z} \rfloor$ and $1 \le j, s_{m+i} < q-z$, we have

$$1 \le s_{m+i} + 1 \le q - z \quad \text{and}$$

$$q - z + 1 \le (\eta'_i - \eta_i)(q-z) + j < q$$

This implies $s_{m+i} + 1 \neq (\eta'_i - \eta_i)(q-z) + j$, a contradiction to (19). That is the subcase $\eta_i > \eta'_i$ is impossible. If $\eta_i > \eta'_i$, (18) can be written as

$$s_{m+i} + (\eta_i - \eta'_i)(q-z) + 1 = j,$$

which is impossible as $q - z + 1 < s_{m+i} + (\eta_i - \eta'_i)(q-z) + 1 < q$ and $1 \le j < q - z$. Hence $p_{a,b'} = *$. Similarly we can prove $p_{a',b} = *$. Then C3-b) holds.

APPENDIX B: PROOF OF THEOREM 2

Proof: Firstly, we will check the parameters K, F, Z and S. Clearly $K = q(m\lfloor \frac{q-1}{q-z} \rfloor + 1)$, $F = q^m$ and $Z = zq^{m-1}$. We now determine the value of S. According to the proof of Theorem 1, $\mathbf{P}$ is an $(mq(\lfloor \frac{q-1}{q-z} \rfloor), q^m, zq^{m-1}, (q-z)q^m)$ PDA with

$$\begin{aligned} \mathcal{S} = \{(s_0, s_1, \ldots, s_m) \mid \\ s_0, s_1, \ldots, s_{m-1} \in [0, q), \ s_m \in [0, q-z)\}. \end{aligned}$$

So we only need to consider the alphabet of $\mathbf{C}$. For any $s = (s_0, s_1, \ldots, s_{m-1}) \in \mathcal{S}$, consider the value of s_m. For any integer $\beta \in [0, q)$, by (4) $p_{a,b}$ is a vector if and only if $\sum_{k=0}^{m-1} \alpha_k \in [0, q) \setminus Y_{\beta,z}$. Then

$$\begin{aligned} s_m \in \{\beta - 1 - c \mid c \in [0, q) \setminus Y_{\beta,z} = \{\beta + z, \\ \beta + z + 1, \ldots, \beta + q - 1\}_q\} \\ = \{q - z - 1, q - z - 2, \ldots, 0\}_q. \end{aligned}$$

And s_m can get every value in $\{0, \ldots, q - z - 2, q - z - 1\}_q$ when $\sum_{k=0}^{m-1} \alpha_k$ goes through $[0, q) \setminus Y_{\beta,z}$. So the alphabet of $\mathbf{C}$ is also $\mathcal{S}$. From the above discussions, we know that $S = (q-z)q^m$.

Next, we will check the properties in Definition 1. First by the above discussions about parameter Z, we have C1 always holds. C2 always holds from the proof of Theorem 1. Now we will prove that C3 holds. Suppose that $p_{a,b}$ and $p_{a',b'}$ are two entries with $p_{a,b} = p_{a',b'} = s \in \mathcal{S}$ where

$$\begin{aligned} &a = (\alpha_0, \alpha_1, \ldots, \alpha_{m-1})_q, \quad a' = (\alpha'_0, \alpha'_1, \ldots, \alpha'_{m-1})_q, \\ &b = (\beta, \gamma, \eta), \quad b' = (\beta', \gamma', \eta'). \end{aligned}$$

We only need to consider the following cases, where all the operations are performed modulo q.

- If $\gamma \in [0, m)$ and $\gamma' = m$, we have $p_{a,b}$ and $p_{a',b'}$ in distinct columns. From (3) and (4) there is an integer $\eta \in [0, \lfloor \frac{q-1}{q-z} \rfloor)$ satisfying

$$\begin{aligned} &\alpha_\gamma - \beta - 1 = \beta' - \sum_{k=0}^{m-1} \alpha'_k - 1, \\ &\alpha'_\gamma = s_\gamma = \beta - \eta(q-z), \quad \alpha_k = \alpha'_k \end{aligned} \tag{20}$$

for any $k \in [0, m) \setminus \{\gamma\}$.

– C3-a) holds. Assume that $a = a'$, which implies that s appears in the row a at least twice. Then $\alpha_\gamma = \alpha'_\gamma$.

From the first and the second items in (20), we have $\alpha_\gamma = \beta + \beta' - \sum_{k=0}^{m-1} \alpha'_k$ and $\alpha'_\gamma = \beta - \eta(q-z)$, respectively. Together with the above fact $\alpha_\gamma = \alpha'_\gamma$, we know that

$$\eta(q-z) + \beta' - \sum_{k=0}^{m-1} \alpha'_k = 0.$$

Since $\sum_{k=0}^{m-1} \alpha'_k \notin Y_{\beta',z}$, then there is an element $j \in \{1, 2, \ldots, q-z\}$ satisfying $\sum_{k=0}^{m-1} \alpha'_k = \beta' - j$. Hence

$$\eta(q-z) + \beta' - (\beta' - j) = \eta(q-z) + j = 0,$$

which is impossible as

$$\begin{aligned} 1 &\leq j + \eta(q-z) \\ &\leq (q-z) + (\frac{q-1}{q-z} - 1)(q-z) \\ &= q-1 \end{aligned} \tag{21}$$

by the fact $\eta \in [0, \lfloor \frac{q-1}{q-z} \rfloor)$.

- C3-b) holds. If $p_{a',b} \neq *$, we have $\alpha'_\gamma \notin X_{\beta,z}$ by (3). Hence there is an element $j \in \{1, 2, \ldots, q-z\}$ satisfying $\alpha'_\gamma = \beta - j$. Submitting $\alpha'_\gamma = \beta - j$ into the second item in (20) we have $\eta(q-z) + j = 0$, wihch is impossible from (21). If $p_{a,b'}$ is an integer, we have $\sum_{k=0}^{m-1} \alpha_k \in \{\beta' - (q-z), \beta' - (q-z-1), \ldots, \beta' - 1\}_q$. Then there is an element $j \in \{1, 2, \ldots q-z\}$ satisfying $\sum_{k=0}^{m-1} \alpha_k = \beta' - j$. From (20), we have

$$\begin{aligned} \sum_{k=0}^{m-1} \alpha_k &= \sum_{k \in [0,m) \setminus \{\gamma\}} \alpha_k + \alpha_\gamma \\ &= \sum_{k \in [0,m) \setminus \{\gamma\}} \alpha_k + (\beta + \beta' - \sum_{k=0}^{m-1} \alpha'_k) \\ &= \sum_{k \in [0,m) \setminus \{\gamma\}} \alpha_k + (\beta + \beta' \\ &\quad - \sum_{k \in [0,m) \setminus \{\gamma\}} \alpha'_k - \alpha'_\gamma) \\ &= \beta + \beta' - \alpha'_\gamma \\ &= \beta + \beta' - (\beta - \eta(q-z)) \\ &= \beta' + \eta(q-z) \end{aligned}$$

Then $\beta' - j = \beta' + \eta(q-z)$. Hence $j + \eta(q-z) = 0$, which is impossible from (21).

- If $\gamma = \gamma' = m$, from (4) we have

$$\beta - \sum_{k=0}^{m-1} \alpha_k - 1 = \beta - \sum_{k=0}^{m-1} \alpha'_k - 1, \quad \alpha_k = \alpha'_k$$

for any $k \in [0, m)$. Then we know that $a = a'$ and $b = b'$, a contradiction to our hypothesis.

REFERENCES

[1] M. M. Amiri and D. Gunduz, "Fundamental limits of coded caching: Improved delivery rate-cache capacity tradeoff," *IEEE Trans. Commun.*, vol. 65, no. 2, pp. 806–815, Feb. 2017.

[2] M. Cheng, Q. Yan, X. Tang, and J. Jiang, "Coded caching schemes with low rate and subpacketizations," 2017, *arXiv:1703.01548*. [Online]. Available: http://arxiv.org/abs/1703.01548

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

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

[5] H. H. Suthan Chittoor, M. Bhavana, and P. Krishnan, "Coded caching via projective geometry: A new low subpacketization scheme," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Jul. 2019, pp. 682–686.

[6] H. Hara Suthan Chittoor, P. Krishnan, K. V. Sushena Sree, and M. V. N. Bhavana,"Subexponential and linear subpacketization coded caching via line graphs and projective geometry," 2020, *arXiv:2001.00399*. [Online]. Available: http://arxiv.org/abs/2001.00399

[7] H. Ghasemi and A. Ramamoorthy, "Improved lower bounds for coded caching," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Hong Kong, Jun. 2015, pp. 1696–1700.

[8] M. Gregori, J. Gomez-Vilardebo, J. Matamoros, and D. Gunduz, "Wireless content caching for small cell and D2D networks," *IEEE J. Sel. Areas Commun.*, vol. 34, no. 5, pp. 1222–1234, May 2016.

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

[10] M. Ji, A. M. Tulino, J. Llorca, and G. Caire, "Order-optimal rate of caching and coded multicasting with random demands," in *Proc. 11th ISWCS*, Aug. 2014, pp. 922–926.

[11] P. Krishnan, "Coded caching via line graphs of bipartite graphs," in *Proc. IEEE Inf. Theory Workshop (ITW)*, Nov. 2018, pp. 1–5.

[12] S. Jin, Y. Cui, H. Liu, and G. Caire, "Uncoded placement optimization for coded delivery," in *Proc. 16th Int. Symp. Model. Optim. Mobile, Ad Hoc, Wireless Netw. (WiOpt)*, May 2018, pp. 1–8.

[13] E. Lampiris and P. Elia, "Adding transmitters dramatically boosts coded-caching gains for finite file sizes," 2018, *arXiv:1802.03389*. [Online]. Available: http://arxiv.org/abs/1802.03389

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

[15] J. Michel and Q. Wang, "Placement delivery arrays from combinations of strong edge colorings," 2019, *arXiv:1907.03177*. [Online]. Available: http://arxiv.org/abs/1907.03177

[16] R. Pedarsani, M. A. Maddah-Ali, and U. Niesen, "Online coded caching," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Jun. 2014, pp. 1878–1883.

[17] A. Sengupta, R. Tandon, and T. C. Clancy, "Improved approximation of storage-rate tradeoff for caching via new outer bounds," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Jun. 2015, pp. 1691–1695.

[18] 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.

[19] 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.

[20] 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)*, Jun. 2017, pp. 1237–1241.

[21] K. Shanmugam, A. G. Dimakis, J. Llorca, and A. M. Tulino, "A unified Ruzsa-Szemerédi framework for finite-length coded caching," in *Proc. 51st Asilomar Conf. Signals, Syst., Comput.*, Oct. 2017, pp. 631–635.

[22] W. Song, K. Cai, and L. Shi, "Some new constructions of coded caching schemes with reduced subpacketization," 2019, *arXiv:1908.06570*. [Online]. Available: http://arxiv.org/abs/1908.06570

[23] C. Tian and J. Chen, "Caching and delivery via interference elimination," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Jul. 2016, pp. 830–834.

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

[25] C.-Y. Wang, S. H. Lim, and M. Gastpar, "A new converse bound for coded caching," in *Proc. IEEE ITW*, Oct. 2016, pp. 1–6.

[26] 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, Jul. 2017.

[27] 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.

[28] Q. Yu, M. A. Maddah-Ali, and A. S. Avestimehr, "The exact rate-memory tradeoff for caching with uncoded prefetching," *IEEE Trans. Inf. Theory*, vol. 64, no. 2, pp. 1281–1296, Feb. 2018.

MINQUAN CHENG received the Ph.D. degree from the Department of Social Systems and Management, Graduate School of Systems and Information Engineering, University of Tsukuba, Tsukuba, Ibaraki, Japan, in 2012. He joined Guangxi Normal University, Guilin, Guangxi, China, where he is currently a Full Professor with the School of Computer Science and Information Technology. His research interests include combinatorics, coding theory, cryptography, and their interactions.

WENYU ZHANG received the bachelor's degree in science and technology of computer from the Technology and Engineering College, Shenyang University, Shenyang, China, in 2016. He is currently pursuing the M.S. degree in computer science and information technology with Guangxi Normal University, Guilin, China. His research interests include combinatorics and coding theory.

JING JIANG received the Ph.D. degree from the Department of Social Systems and Management, Graduate School of Systems and Information Engineering, University of Tsukuba, Tsukuba, Ibaraki, Japan, in 2015. He is currently a Lecturer with the School of Computer Science and Information Technology, Guangxi Normal University, Guilin, Guangxi, China. His research interests include combinatorial design theory, coding theory, and their interactions.