A New Construction Structure on Coded Caching with Linear Subpacketization: Non-Half-Sum Disjoint Packing

Minquan Cheng, Huimei Wei, Kai Wan, and Giuseppe Caire

Abstract

Coded caching is a promising technique to effectively reduce peak traffic by using local caches and the multicast gains generated by these local caches. We prefer to design a coded caching scheme with the subpacketization F and transmission load R as small as possible since these are the key metrics for evaluating the implementation complexity and transmission efficiency of the scheme, respectively. However, most of the existing coded caching schemes have large subpacketizations which grow exponentially with the number of users K, and there are a few schemes with linear subpacketizations which have large transmission loads. In this paper, we focus on studying the linear subpacketization, i.e., $K = F$, coded caching scheme with low transmission load. Specifically, we first introduce a new combinatorial structure called non-half-sum disjoint packing (NHSDP) which can be used to generate a coded caching scheme with $K = F$. Then a class of new schemes is obtained by constructing NHSDP. Theoretical and numerical comparisons show that (i) compared to the existing schemes with linear subpacketization (to the number of users), the proposed scheme achieves a lower load; (ii) compared to some existing schemes with polynomial subpacketization, the proposed scheme can also achieve a lower load in some cases; (iii) compared to some existing schemes with exponential subpacketization, the proposed scheme has loads close to those of these schemes in some cases. Moreover, the new concept of NHSDP is closely related to the classical combinatorial structures such as cyclic difference packing (CDP), non-three-term arithmetic progressions (NTAP), and perfect hash family (PHF). These connections indicate that NHSDP is an important combinatorial structure in the field of combinatorial design.

Index Terms

Coded caching scheme, placement delivery array, linear subpacketization, transmission load, non-half-sum disjoint packing

I. INTRODUCTION

The rise of wireless devices in recent years has significantly increased network traffic, fueled by activities such as multimedia streaming, web browsing, and social networking. In addition, the high temporal variability of this traffic leads to congestion during peak periods and inefficient use of network resources during off-peak times. Caching is a promising technique to reduce peak traffic by taking advantage of memories distributed across the network to duplicate content during off-peak times. Conventional uncoded caching techniques focus on predicting the user demands for making an appropriate prefetching strategy, thus realizing a "local caching gain", which scales with the amount of local memory [1]. In their seminal paper [2], Maddah-Ali and Niesen (MN) demonstrated that, in addition to a local caching gain, coded caching can also attain a "global caching gain". This global caching gain scales with the global amount of memory in the network, since each transmission of the MN scheme can serve multiple users simultaneously.

In a (K, M, N) coded caching system, there is a central server having N equal-sized files and K users each of whom can cache up to M files. The server connects to the users over an error-free shared link. An F-division (K, M, N) coded caching scheme consists of two phases: the placement phase during the off-peak hours and the delivery phase during the peak hours. In the placement phase, the server divides each file into F equal packets and places some of these packets into each user's cache without knowledge of the users' future demands. If packets are directly placed in each user's cache, it is referred to as uncoded placement. Otherwise it is called coded placement. The parameter F is called the subpacketization. In the delivery phase, each user requests one file randomly. According to the users' demands and the caching packets, the server broadcasts some coded packets such that each user can recover its desired file. The normalized amount of transmission for the worst-case over all possible demands is called the transmission load (or load) R.

The first well-known coded caching scheme was proposed by Maddah-Ali and Niesen in [2], which utilized combinatorial design in the placement phase and linear coding in the delivery phase. This scheme is referred to as the MN scheme. The authors

M. Cheng and H. Wei are with Guangxi Key Lab of Multi-source Information Mining & Security, Guangxi Normal University, Guilin 541004, China (e-mail: chengqinshi@hotmail.com, scarlett_w_edu@163.com).

K. Wan is with the School of Electronic Information and Communications, Huazhong University of Science and Technology, 430074 Wuhan, China, (e-mail: kai_wan@hust.edu.cn).

G. Caire is with the Electrical Engineering and Computer Science Department, Technische Universität Berlin,10587 Berlin, Germany (e-mail: caire@tu-berlin.de). The work of G. Caire was partially funded by the European Research Council under the ERC Advanced Grant N. 789190, CARENET.

in [3]–[6] separately proved the minimality of the load of the MN scheme under the uncoded placement and for the case $K \leq N$. Using the placement strategy of the MN scheme, the authors in [5] proposed an exact optimal caching scheme, which strictly improves upon the current state of the art by exploiting commonalities among user demands under uncoded placement.

However, the subpacketization $F = \binom{K}{KM/N}$ of the MN scheme increases exponentially with the number of users K. This would become infeasible for practical implementation when K is large. So it is meaningful to design a coded caching scheme with low subpacketization. The grouping method, which is widely regarded as the most effective in reducing subpacketization, was proposed in [7], [8]. Nevertheless, the load of the scheme generated by grouping method increases fast. Recently, the authors in [9] proposed an interesting combinatorial structure called placement delivery array (PDA) to study the coded caching scheme with low subpacketization. They also showed that the MN scheme is equivalent to a special PDA which is called MN PDA. Based on the concept of PDA, there are many works focusing on achieving the low subpacketization, such as [8]–[20].

There are other characterizations of coded caching schemes, such as the linear block codes [21], the special $(6, 3)$-free hypergraphs [22], the (r, t) Ruzsa-Szeméredi graphs [23], the strong edge coloring of bipartite graphs [24], cross resolvable designs [25], the projective geometry [26], and combinatorial designs [27] which are listed in Table I. Here we do not include the schemes from in [23] and the first scheme in [28] in the Table I, since we focus on explicit constructions of schemes, while theirs only focus on the existing and the user number approximates infinite integer.

By Table I, we can see that the schemes in [2], [8]–[10], [12], [13], [19], [21], [22] have flexible number of users, large memory regimes, and small loads, but large subpacketizations which increase exponentially (or sub-exponentially) with the number of users; the schemes in [14]–[16], [20], [24], [26]–[30] have low subpacketizations which increase polynomially or linearly with the number of users. However, the schemes in [15], [24], [26] have the special numbers of users (i.e., combinations, powers, or products of combinations and powers) and special memory ratio (i.e., the ratios of these combinations, powers, or products of combinations and powers); the schemes in [16], [27], [28] have the large memory ratio which approximates 1; the schemes in [20], [26] have the memory ratios approximating 0 or 1. The schemes in [14], [29] have the subpacketizations which increase linearly with the number of users for the flexible number of users and large memory regimes. It is worth noting that the schemes in [14], [29] use the same placement strategy, consecutive cyclic uncoded placement. The authors showed that under the consecutive cyclic uncoded placement and one-shot delivery, the maximum coded caching gain is $2\lfloor \frac{K}{K-KM/N+1} \rfloor + 1$. Clearly when M/N is small such that $M/N \leq 1/2$, the linear coded caching schemes have the maximum coded caching gain 3.

A. Contribution

In this paper, we focus on constructing the coded caching scheme with linear subpacketization. When K is odd, we introduce a new concept called Non-half-sum disjoint packing (NHSDP) which can realize a coded caching scheme with linear subpacketization. Compared to the existing characterizing method, the greatest advantage of our NHSDP is that it integrates the placement strategy and the transmission strategy into a single condition, i.e., the second condition of NHSDP in Definition 2. The main result can be summarized as follows.

- When K is odd, we transform the coded caching construction to a new combinatorial structure named Non-half-sum disjoint packing (NHSDP). Given a (v, g, b) NHSDP for any odd positive integers v, g, and b, we can obtain a $(K = v, M, N)$ coded caching scheme with the memory ratio $\frac{M}{N} = 1 - \frac{bg}{v}$, the subpacketization $F = K$, and the transmission load $R = b$. For the even number of users K, we can add one virtual user into the system such that the effective number of users is $K' = K + 1$, which could be solved by the $(K' + 1, g, b)$ NHSDP.
- By constructing an NHSDP, we can obtain a $(K = v, M, N)$ coded caching scheme with the memory ratio $M/N = 1 - \frac{2^n \prod_{i=1}^{n} m_i}{v}$, the coded caching gain $g = 2^n$, and the transmission load $R = \prod_{i=1}^{n} m_i$ for any odd integer $v \geq 2\phi(m_1, m_2, \ldots, m_n) + 1$ where n and m_1, m_2, …, m_n are any positive integers. In particular, when $m_1 = \cdots = m_n = \lfloor \frac{v^{1/n}-1}{2} \rfloor$, we obtain a $(K = v, M, N)$ coded caching scheme with the memory ratio $\frac{M}{N} = 1-(\frac{2\lfloor \frac{q-1}{2} \rfloor}{q})^n$, the subpacketzation $F = K$, the coded caching gain $g = 2^n$, and the transmission load $R = \lfloor \frac{q-1}{2} \rfloor^n$.
- Theoretical and numerical comparisons show that the proposed scheme achieves a lower load compared to the existing schemes with linear subpacketization (to the number of users); compared to some existing schemes with polynomial subpacketization, our proposed scheme can achieve a lower transmission load and also has lower load in some cases; compared to some existing schemes with exponential subpacketization, our scheme has loads close to those of these schemes in some cases.
- The new concept of NHSDP has a close relationship between the classic combinatorial structures, such as cyclic difference packing (CDP) [31], non-three-term arithmetic progressions (NTAP) [32], and perfect hash family (PHF) [33], etc. Specifically, a CDP can be used to construct a $(v, g, 1)$ NHSDP. Given a $(v, g, 1)$ NHSDP is equivalent to an NTAP set over $[v]$ and it can be used to construct a $(3 : gv, v, 3)$ PHF. In cryptography [33], combinatorics [34], and coding theory [35], we aim to construct a subset of $\mathbb{Z}_n$ satisfying NTAP, with the maximum cardinality, and a $(3 : m, v, 3)$ PHF with the maximum value of m. When $v = 3^n$, we can obtain a class of NTAPs using our new NHSDP whose size is larger than

TABLE I: The existing coded caching schemes where $K, k, t, m, H, a, z, r \in \mathbb{Z}^+$, $\begin{bmatrix} k \\ t \end{bmatrix}_q = \frac{(q^k-1)\ldots(q^{k-t+1}-1)}{(q^t-1)\ldots(q-1)}$, $\langle K \rangle_t = K \mod t$.

Schemes & References	Number of Users K	Memory ratio	Load	Subpacketization	Constrain
MN Scheme [2]	K	$\frac{t}{K}$	$\frac{K-t}{t+1}$	$\binom{K}{t}$	$t \leq K$
WCLC scheme [12], [19], [22]	$\binom{m}{z}k^z$	$1-\left(\frac{k-t}{k}\right)^z$	$(\frac{k-t}{\lfloor \frac{k-1}{k-t} \rfloor})^z$	$\lfloor \frac{k-1}{k-t} \rfloor^z k^{m-1}$	$1 \leq t < k,$ $1 \leq z \leq m$
YTCC scheme [24]	$\binom{H}{a}$	$1-\frac{\binom{a}{r}\binom{H-a}{z-r}}{\binom{H}{z}}$	$\frac{\binom{H}{a+z-2r}}{\binom{H}{z}} \cdot \min\{\binom{H-a-z+2r}{a-r}, \binom{a+z-2r}{a-r}\}$	$\binom{H}{z}$	$r < a < H,$ $r < z < H,$ $a+z \leq H+r$
WCCLS scheme [15]	q^m	$1-\frac{\binom{m}{w}(q-1)^w}{q^m}$	$\frac{\binom{m}{w}(q-1)^w}{q^{m-w}}$	q^m	$m, w \in \mathbb{Z}^+, m < w$
CKSM scheme 1 [26]	$\frac{1}{t!}q^{\frac{t(t-1)}{2}} \cdot \prod_{i=0}^{t-1} \begin{bmatrix} k-i \\ 1 \end{bmatrix}_q$	$1-q^{mt} \cdot \prod_{i=0}^{m-1} \frac{\begin{bmatrix} k-t-i \\ 1 \end{bmatrix}_q}{\begin{bmatrix} k-i \\ 1 \end{bmatrix}_q}$	$\frac{m!q^{mt}}{(m+t)!}q^{\frac{t(t-1)}{2}} \cdot \prod_{i=0}^{t-1} \begin{bmatrix} k-m-i \\ 1 \end{bmatrix}_q$	$\frac{1}{m!}q^{\frac{m(m-1)}{2}} \cdot \prod_{i=0}^{m-1} \begin{bmatrix} k-i \\ 1 \end{bmatrix}_q$	$m+t \leq k,$ prime power
CKSM scheme 2 [26]	$\begin{bmatrix} k \\ t \end{bmatrix}_q$	$1-\frac{\begin{bmatrix} k-t \\ m \end{bmatrix}_q}{\begin{bmatrix} k \\ m+t \end{bmatrix}_q}$	$\frac{\begin{bmatrix} k \\ m \end{bmatrix}_q}{\begin{bmatrix} k \\ m+t \end{bmatrix}_q}$	$\begin{bmatrix} k \\ m+t \end{bmatrix}_q$	$2 \leq q$
ASK scheme 1 [27]	q^2+q+1	$\frac{q^2}{q^2+q+1}$	1	q^2+q+1	prime power
ASK scheme 2 [27]	q^2	$\frac{q-1}{q}$	$\frac{q}{q+1}$	q^2+q	$2 \leq q$
ZCW scheme [16]	2^m	$1-\frac{\binom{m}{\omega}}{\sum_{i=0}^{\omega}\binom{m}{i}}$	$\frac{\binom{m}{\omega}2^{m-\omega}}{\sum_{i=0}^{\omega}\binom{m}{i}}$	$\sum_{i=0}^{\omega}\binom{m}{i}$	$\omega < m$
WCWL scheme [14]	K	$\frac{t}{K}$	$\frac{(K-t)(K-t+1)}{2K}$	K	$0 \leq t \leq K,$ $(K-t+1) \vert K$ or $K-t=1$
			$\frac{K-t}{2\lfloor \frac{K}{K-t+1} \rfloor+1}$	$\left(2\lfloor \frac{K}{K-t+1} \rfloor+1\right)K$	$0 \leq t \leq K,$ $\langle K \rangle_{K-t+1} = K-t$
			$\frac{K-t}{2\left\lfloor \frac{K}{K-t+1} \right\rfloor}$	$2\left\lfloor \frac{K}{K-t+1} \right\rfloor K$	$0 \leq t \leq K,$ otherwise
XXGL scheme [28]	K	$\frac{K-2}{K}$	$\frac{K-1}{K}$	K	$K \in \mathbb{N}^+$
AST scheme [20]	$2^r k$	$1-\frac{r+1}{2^r}+\frac{r}{2^r k}$	$\frac{k(r+1)-r}{2^r}$	$2^r k = K$	$r, k \in \mathbb{N}^+$
MR scheme [29]	K	$\frac{t}{K}$	$\left\lceil \frac{K(K-t)}{2+\lfloor \frac{t}{K-t+1} \rfloor+\lfloor \frac{t-1}{K-t+1} \rfloor} \right\rceil \cdot \frac{1}{K}$	K	$K, t \in \mathbb{Z}^+, t < K$

the state-of-the-art achievable bound on the NTAP size in [36] when $n \leq 52$. The PHF derived from our proposed NHSDP has more columns compared to the first quadrics PHF [37]. Furthermore, the number of columns in our proposed PHF is close to that of the Hermitian PHF presented in [37].

B. Organizations and notations

The rest of this paper is organized as follows. In Section II, a coded caching caching system, concept of PDA and their relationship are introduced. In Section III, we introduce the concept of the non-half-sum disjoint packing (NHSDP) and show that it can be used to generate a PDA. In Section IV, we propose a new class of PDAs by constructing an NHSDP. In Section V, we provide the performance analysis of the proposed scheme and we expand the application of NHSDP in Section VI. Finally, we conclude this paper in Section VII.

Notation: In this paper, we will use the following notations. Let bold capital letter, bold lowercase letter and curlicue letter denote array, vector and set respectively; let $|\mathcal{A}|$ denote the cardinality of the set $\mathcal{A}$; define $[a] = \{1, 2, \ldots, a\}$ and $[a:b]$ is the

set $\{a, a+1, \ldots, b-1, b+1\}$; $\lfloor a \rfloor$ denotes the largest integer not greater than a. $a \nmid b$ denotes a does not divide b. Define that $\begin{bmatrix} k \\ t \end{bmatrix}_q = \frac{(q^k-1)\ldots(q^{k-t+1}-1)}{(q^t-1)\ldots(q-1)}$, $\langle K \rangle_t = K \mod t$; $\mathbb{Z}_v$ is a positive integer ring.

II. PRELIMINARIES

In this section, we will introduce a coded caching system, placement delivery array and their relationship respectively.

A. Coded caching system

A (K, M, N) coded caching system illustrated in Figure 1 contains a server storing N equal-sized files $\mathcal{W} = \{W_n \mid n \in [N]\}$, K users each of which can cache at most M files where $0 \leq M \leq N$. The sever connects the users over an error-free shared-link. An F-division (K, M, N) coded caching scheme operates in two phases:

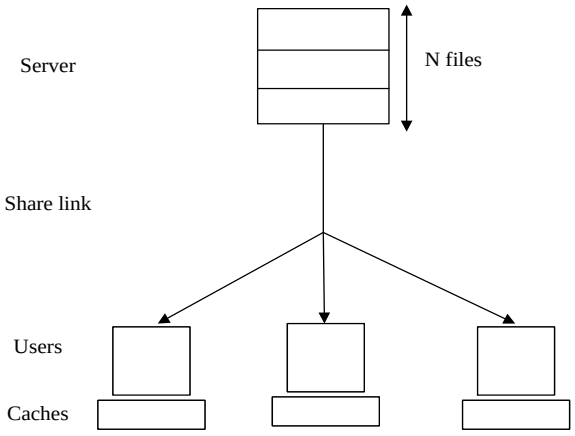

Fig. 1: (K, M, N) caching system

- **Placement phase:** The server divides each file into F equal sized packets, i.e., $W_n = \{W_{n,j} \mid j \in [F]\}$ where $n \in [N]$, and places some packets into each user's cache without knowing any information about future demands. Let $\mathcal{Z}_k$ be the packets cached by user k. This placement strategy is called uncoded placement, otherwise it is called coded placement.
- **Delivery phase:** Each user requests one file from $\mathcal{W}$ randomly. Denote the requested file number by $\mathbf{d}$=$(d_1, d_2, \ldots, d_K)$, i.e., user k requests file W_{d_k}, where $k \in \mathcal{K}, d_k \in [N]$. After receiving the request vector $\mathbf{d}$, the sever broadcasts XOR of packets with a size of at most $R_{\mathbf{d}}F$ to the users such that each user is able to decode its requested file.

In this paper, we focus on the normalized amount of transmission for the worst-case over all possible demands which is defined as follows.

$$R = \max\{R_{\mathbf{d}} \mid \mathbf{d} \in [N]^K\} \tag{1}$$

The first well-known scheme was proposed in [2], which is referred to as the MN scheme, has the minimum load under uncoded placement [3]–[6] but has large subpacketization which increases exponentially with the number of users.

B. Placement delivery array

In order to study the scheme with lower subpacketization, the authors in [9] proposed an interesting combinatorial structure called placement delivery array defined as follows.

Definition 1 ([9])**.** For positive integers K, F, Z and S, an $F \times K$ array $\mathbf{P} = (p_{j,k})$, where $j \in [F]$ and $k \in [K]$, composed of " $*$ " called star and $[S]$, is called a (K, F, Z, S) placement delivery array (PDA) if the following conditions hold:

C1. Each column has exactly Z stars.
C2. Each integer in $[S]$ occurs at least once.
C3. For any two distinct entries p_{j_1,k_1} and p_{j_2,k_2}, $p_{j_1,k_1} = p_{j_2,k_2} = s$ if only if
   a. $j_1 \neq j_2, k_1 \neq k_2$, i.e., they lie in distinct rows and distinct columns;
   b. $p_{j_1,k_2} = p_{j_2,k_1} = *$, i.e., the corresponding 2×2 subarray formed by rows j_1, j_2 and columns k_1, k_2 must be one of the following form,

$$\begin{pmatrix} s & * \\ * & s \end{pmatrix} \quad \text{or} \quad \begin{pmatrix} * & s \\ s & * \end{pmatrix}. \tag{2}$$

A PDA is called g-regular if each integer occurs exactly g times. One can check that the following array $\mathbf{P}$ is a 2-$(4,4,2,4)$ PDA.

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

By Algorithm 1, for any given PDA, we can obtain the following result.

Lemma 1. ([9]) Given a (K,F,Z,S) PDA, there exists an F-division coded caching scheme for the (K,M,N) coded caching system with memory ratio $\frac{M}{N}=\frac{Z}{F}$, subpacketization F, and load $R=\frac{S}{F}$. □

Algorithm 1 Caching scheme based on PDA in [9]

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

For example, using Algorithm 1, a 4-division $(4,2,4)$ coded caching scheme based on $\mathbf{P}$ in (3) can be obtained as follows.

- **Placement phase**: From Line 2, we split each file into 4 packets, i.e., $W_n=\{W_{n,j} \mid n\in[4], j\in[4]\}$. By Lines 3-5, the caches for all of users are

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

  Clearly each user caches exactly $2\times 4=8$ packets, i.e., $M=2$ files. So we have $M/N=2/4=1/2$, i.e., $M/N=Z/F$.
- **Delivery phase**: Assume that the request vector is $\mathbf{d}=(1,2,3,4)$. By Lines 8-10, the server transmits $W_{1,2}\oplus W_{2,1}$ at the first time slot, $W_{2,3}\oplus W_{3,2}$ at the second time slot, $W_{3,4}\oplus W_{4,3}$ at the third time slot, and $W_{1,4}\oplus W_{4,1}$ at the last time slot. It is not difficult to check that each user can decode its requested file. There are exactly 4 coded signals transmitted by the server. From (1) we have $R=4/4=1$, i.e., $R=S/F$.

There are many constructions based on PDA [8]–[12], [16], [22], [24], [38], [39]. The authors in [9] showed that the MN scheme can also be represented by a PDA. Some other constructions could be also represented by appropriate PDAs such as the caching schemes based on liner block code [21], projective space [26], and combinatorial designs [27], etc. We list their performances in Table I. In addition, given a PDA, we can obtain a PDA with any number of users by the following result.

Lemma 2 (Grouping method [8])**.** Given a (K_1,F,Z,S) PDA, there exists a (K,h_1F,h_1Z,hS) PDA for any $K>K_1$ where $h_1=\frac{K_1}{gcd(K_1,K)}$ and $h=\frac{K}{gcd(K_1,K)}$.

III. NON-HALF-SUM DISJOINT PACKING

Recall that Condition C3-a) of Definition 1 is called the Latin property in combinatorial design theory. Naturally, we can construct a PDA with $K=F$ by first constructing a Latin square and then putting stars in some integer entries such that the resulting array satisfies Conditions C1, C2, and C3. Then we obtain the desired PDA with linear subpacketization. Specifically, we use the classical and simple construction of the Latin square $\mathbf{L}=(l_{f,k})_{f,k\in\mathbb{Z}_v}$ where the entry $l_{f,k}=f+k$ for each f, $k\in\mathbb{Z}_v$. Recall that $\mathbb{Z}_v$ is a positive integer ring, i.e., the arithmetic operations in the ring $\mathbb{Z}_v$. When we cyclically replace the integer entries by stars, Condition C3 of Definition 1 is transformed into the following novel combinatorial structure for the integer entries of the first row of the Latin square.

Definition 2 (Non-half-sum disjoint packing, NHSDP)**.** For any positive odd integer v, a pair $(\mathbb{Z}_v,\mathfrak{D})$ where $\mathfrak{D}$ consists of b g-subsets of $\mathbb{Z}_v$ is called (v,g,b) non-half-sum disjoint packing if it satisfies the following conditions.

- The intersection of any two different elements in $\mathfrak{D}$ is empty;
- For each $\mathcal{D} \in \mathfrak{D}$, the half-sum of any two different elements in $\mathcal{D}$ (i.e., the sum of the two elements divided by 2) does not appear in any block of $\mathfrak{D}$.[1]

□

Let us take the following example with $v = 15$ to further explain the concept of NHSDP.

Example 1. Consider the following $(v = 15, g = 4, b = 2)$ NHSDP

$$\mathfrak{D} = \{\mathcal{D}_1 = \{-1, 1, -2, 2\} = \{14, 1, 13, 2\},\ \mathcal{D}_2 = \{-4, 4, -5, 5\} = \{11, 4, 10, 5\}\}. \tag{4}$$

Clearly $\mathcal{D}_1 \cap \mathcal{D}_2 = \emptyset$, i.e., the first condition of Definition 2 holds. The half-sums of any two different elements in each block in $\mathfrak{D}$ are as follows,

$$\begin{aligned} \text{in } \mathcal{D}_1 &: \frac{-1+1}{2} = 0,\ \frac{-1-2}{2} = 6,\ \frac{-1+2}{2} = -7,\ \frac{1-2}{2} = 7,\ \frac{1+2}{2} = -6,\ \frac{-2+2}{2} = 0, \\ \text{in } \mathcal{D}_2 &: \frac{-4+4}{2} = 0,\ \frac{-4-5}{2} = 3,\ \frac{-4+5}{2} = -7,\ \frac{4-5}{2} = 7,\ \frac{4+5}{2} = -3,\ \frac{-5+5}{2} = 0. \end{aligned} \tag{5}$$

We have $\{0, \pm 3, \pm 6, \pm 7\} \cap \mathcal{D}_1 = \{0, \pm 3, \pm 6, \pm 7\} \cap \mathcal{D}_2 = \emptyset$, i.e., the second condition of Definition 2 holds. So $(\mathbb{Z}_{15}, \mathfrak{D})$ is a $(15, 4, 2)$ NHSDP. □

Using an NHSDP, we can obtain a PDA with $F = K$ by the following novel construction.

Construction 1. Given a (v, g, b) NHSDP $(\mathbb{Z}_v, \mathfrak{D})$, then a $v \times v$ array $\mathbf{P} = (p_{f,k})_{f,k \in \mathbb{Z}_v}$ is defined in the following way

$$p_{f,k} = \begin{cases} (f + k, i), & \text{if } k - f \in \mathcal{D}_i,\ \exists\ i \in [b]; \\ *, & \text{otherwise.} \end{cases} \tag{6}$$

□

Now let us take the NHSDP in Example 1 to further illustrate Construction 1.

Example 2. When $v = 15$, we have a $(15, 4, 2)$ NHSDP in Example 1. By Construction 1, the following array can be obtained,

$$\mathbf{P} = \begin{pmatrix} * & (1,1) & (2,1) & * & (4,2) & (5,2) & * & * & * & * & (10,2) & (11,2) & * & (13,1) & (14,1) \\ (1,1) & * & (3,1) & (4,1) & * & (6,2) & (7,2) & * & * & * & * & (12,2) & (13,2) & * & (0,1) \\ (2,1) & (3,1) & * & (5,1) & (6,1) & * & (8,2) & (9,2) & * & * & * & * & (14,2) & (0,2) & * \\ * & (4,1) & (5,1) & * & (7,1) & (8,1) & * & (10,2) & (11,2) & * & * & * & * & (1,2) & (2,2) \\ (4,2) & * & (6,1) & (7,1) & * & (9,1) & (10,1) & * & (12,2) & (13,2) & * & * & * & * & (3,2) \\ (5,2) & (6,2) & * & (8,1) & (9,1) & * & (11,1) & (12,1) & * & (14,2) & (0,2) & * & * & * & * \\ * & (7,2) & (8,2) & * & (10,1) & (11,1) & * & (13,1) & (14,1) & * & (1,2) & (2,2) & * & * & * \\ * & * & (9,2) & (10,2) & * & (12,1) & (13,1) & * & (0,1) & (1,1) & * & (3,2) & (4,2) & * & * \\ * & * & * & (11,2) & (12,2) & * & (14,1) & (0,1) & * & (2,1) & (3,1) & * & (5,2) & (6,2) & * \\ * & * & * & * & (13,2) & (14,2) & * & (1,1) & (2,1) & * & (4,1) & (5,1) & * & (7,2) & (8,2) \\ (10,2) & * & * & * & * & (0,2) & (1,2) & * & (3,1) & (4,1) & * & (6,1) & (7,1) & * & (9,2) \\ (11,2) & (12,2) & * & * & * & * & (2,2) & (3,2) & * & (5,1) & (6,1) & * & (8,1) & (9,1) & * \\ * & (13,2) & (14,2) & * & * & * & * & (4,2) & (5,2) & * & (7,1) & (8,1) & * & (10,1) & (11,1) \\ (13,1) & * & (0,2) & (1,2) & * & * & * & * & (6,2) & (7,2) & * & (9,1) & (10,1) & * & (12,1) \\ (14,1) & (0,1) & * & (2,2) & (3,2) & * & * & * & * & (8,2) & (9,2) & * & (11,1) & (12,1) & * \end{pmatrix}. \tag{7}$$

It is not difficult to check that each column of $\mathbf{P}$ has exactly $Z = v - bg = 15 - 2 \times 4 = 7$ stars, there are exactly $S = 30$ vectors occur in $\mathbf{P}$, and each vector occurs at most once in each row and each column. So the conditions C1, C2, and C3-a) of Definition 1 hold. Finally, let us consider Condition C3-b). Let us first consider the entries $p_{1,0} = p_{0,1} = (1, 1)$. We have $p_{1,1} = p_{0,0} = *$ which satisfies Condition C3-b). The reason for this is as follows. When $f = 1$, if $k = 0$ we have $k - f = -1 = 14 \in \mathcal{D}_1$ in (4) and $1 + 0 = 1$, then we set $p_{1,0} = (1, 1)$; if $k = 1$, we have $k - f = 0$, which does not appear in either $\mathcal{D}_1$ or $\mathcal{D}_2$. Thus we set $p_{1,1} = *$. Similarly we can check that all the vectors in $\mathbf{P}$ satisfy Condition C3-b) of Definition 1. So $\mathbf{P}$ is a $(15, 15, 7, 30)$ PDA which can realize a coded caching scheme with the memory ratio $\frac{M}{N} = \frac{7}{15}$, the subpacketization $F = K = 15$, and load $R = 2$. □

From (6) and Example 2, we can see that Condition C1 of Definition 1 is ensured by the first condition of NHSDP, and Condition C3-b) of Definition 1 is ensured by the second condition of NHSDP. Recall that our construction is based on the

[1]In an NHSDP $(\mathbb{Z}_v, \mathfrak{D})$, each element of $\mathfrak{D}$ is called block.

constructing a Latin square. So the condition C3-a) always holds. Then, for any parameters v, g and b, if there exists a (v, g, b) NHSDP by Construction 1, we can also obtain a PDA in the following theorem, whose proof is given in Appendix A.

Theorem 1 (PDA via NHSDP)**.** Given a (v, g, b) NHSDP, we can obtain a $(v, v, v-bg, bv)$ PDA which realizes a $(K = v, M, N)$ coded caching scheme with memory ratio $\frac{M}{N} = 1 - \frac{bg}{v}$, coded caching gain g, subpacketization $F = v$, and transmission load $R = b$. □

Remark 1. When K is even, we can add a virtual user into the coded caching system, making the efficient number of users $K + 1$ which can be solved by the $(K + 1, g, b)$ NHSDP in Theorem 1.

IV. Constructions of PDAs via NHSDPs

By Theorem 1, in order to obtain a coded caching scheme with linear subpacketization we should study NHSDPs. In this section we will propose a new construction of NHSDPs.

We first introduce the main idea of our new construction of NHSDPs. By embedding integers into high-dimensional geometric spaces, we aim to transform the NHSDP into studying a geometric problem. Specifically, we will design a subset of $\mathbb{Z}_v$, say $\mathcal{X}$, to construct an NHSDP $(\mathbb{Z}_v, \mathfrak{D})$ such that each integer in $\mathfrak{D}$ can be uniquely represented by $\mathcal{X}$, ensuring that the first condition of Definition 2 holds. Furthermore, the half-sum generated by any two different integers in the same block of $\mathfrak{D}$ can also be represented by $\mathcal{X}$, and the coefficient sets for these two representations are disjoint. This approach can naturally distinguish the elements in $\mathfrak{D}$ from their half-sums perfectly which implies that the second condition of Definition 2 is guaranteed. By finding an appropriate subset $\mathcal{X}$, we obtain the following main construction in this paper.

Construction 2. For any n positive integers m_1, m_2, …, m_n, let $\mathcal{A} := [m_1] \times [m_2] \times \cdots \times [m_n]$. Define the following recursive function

$$\begin{cases} f(1) = m_1, & \text{if } i = 1; \\ f(i) = m_i \left(2 \sum_{j=1}^{i-1} f(j) + 1\right), & \text{if } i \geq 2, \end{cases} \tag{8}$$

and $\mathcal{X} := \{x_i = \frac{f(i)}{m_i} | i \in [n]\}$. We can construct a family $\mathfrak{D} = \{\mathcal{D}_{\mathbf{a}} \mid \mathbf{a} = (a_1, a_2, \ldots, a_n) \in \mathcal{A}\}$ where $\mathcal{D}_{\mathbf{a}}$ is defined as

$$\mathcal{D}_{\mathbf{a}} = \left\{ \alpha_1 a_1 x_1 + \alpha_2 a_2 x_2 + \cdots + \alpha_n a_n x_n \;\middle|\; \alpha_i \in \{-1, 1\}, i \in [n] \right\}, \tag{9}$$

for each vector $\mathbf{a} = (a_1, a_2, \ldots, a_n) \in \mathcal{A}$. By the above construction, $\mathfrak{D}$ has $m_1 \times m_2 \times \cdots \times m_n$ blocks, and each block $\mathcal{D}_{\mathbf{a}}$ has 2^n integers. □

Let us take the following example to illustrate Construction 2.

Example 3. When $n = 3$, $m_1 = m_2 = m_3 = 2$, we have $\mathcal{A} = [2]^3$. From (8), when $i = 1$, 2, and 3 we have

$$f(1) = m_1 = 2, \quad f(2) = m_2(2f(1) + 1) = 2(4 + 1) = 10, \quad f(3) = m_3(2f(1) + 2f(2) + 1) = 2(4 + 20 + 1) = 50,$$

respectively. Then we have the subset

$$\mathcal{X} = \left\{ x_1 = \frac{f_1}{m_1} = \frac{2}{2} = 1, x_2 = \frac{f(2)}{m_2} = \frac{10}{2} = 5, x_3 = \frac{f(3)}{m_3} = \frac{50}{2} = 25 \right\}.$$

Now let us consider $\mathfrak{D}$ in Construction 2. When $\mathbf{a} = (1, 1, 1)$ and $(x_1, x_2, x_3) = (1, 5, 25)$, from (9), then we have

$$\begin{aligned} \mathcal{D}_{(1,1,1)} =& \{\alpha_1 a_1 x_1 + \alpha_2 a_2 x_2 + \alpha_3 a_3 x_3 \mid \alpha_1, \alpha_2, \alpha_3 \in \{-1, 1\}\} \\ =& \{1 \cdot \alpha_1 + 5 \cdot \alpha_2 + 25 \cdot \alpha_3 \mid \alpha_1, \alpha_2, \alpha_3 \in \{-1, 1\}\} \\ =& \{31, 21, 29, 19, -19, -29, -21, -31\}. \end{aligned}$$

For instance, the integer 31 in $\mathcal{D}_{(1,1,1)}$ can be obtained by $1 + 5 + 25 = 31$. It is evident that the number of all the possibilities of α_1, α_2, and α_3 is $2^3 = 8$, so the block $\mathcal{D}_{(1,1,1)}$ contains 8 different integers. Similarly, we can obtain the following blocks.

$$\begin{aligned} \mathfrak{D} = \{&\mathcal{D}_{(1,1,1)} = \{31, 21, 29, 19, -19, -29, -21, -31\}, \quad \mathcal{D}_{(2,1,1)} = \{32, 22, 28, 18, -18, -28, -22, -32\}, \\ &\mathcal{D}_{(1,2,1)} = \{36, 16, 34, 14, -14, -34, -16, -36\}, \quad \mathcal{D}_{(2,2,1)} = \{37, 17, 33, 13, -13, -33, -17, -37\}, \\ &\mathcal{D}_{(1,1,2)} = \{56, 46, 54, 44, -44, -54, -46, -56\}, \quad \mathcal{D}_{(2,1,2)} = \{57, 47, 53, 43, -43, -53, -47, -57\}, \\ &\mathcal{D}_{(1,2,2)} = \{61, 41, 59, 39, -39, -59, -41, -61\}, \quad \mathcal{D}_{(2,2,2)} = \{62, 42, 58, 38, -38, -58, -42, -62\}\}. \end{aligned} \tag{10}$$

By the above equation, $\mathfrak{D}$ has $m_1 m_2 m_3 = 8$ elements $\mathcal{D}_{\mathbf{a}}$. In fact, all the integers in $\mathfrak{D}$ can be uniquely represented by $\mathcal{X}$ and the coefficients in $\{\pm 2, \pm 1\}^3$, and all half-sums of any two integers in the same block would not appear in $\mathfrak{D}$. For instance, the

integer $36 \in \mathcal{D}_{(1,2,1)}$ is uniquely expressed by $\mathcal{X} = \{1, 5, 25\}$ where the coefficients are $(1, 2, 1) \in \{\pm 2, \pm 1\}^3$, and -36 can be uniquely represented by $\mathcal{X} = \{1, 5, 25\}$ where the coefficients are $(-1, -2, -1) \in \{\pm 2, \pm 1\}^3$.

Now let we consider the half-sum of 36 and -36. From (10), we have $\frac{36-36}{2} = (\frac{1-1}{2}) \cdot 0 + (\frac{2-2}{2}) \cdot 5 + (\frac{1-1}{2}) \cdot 25 = 0$, i.e, the coefficients are $(0, 0, 0)$ which are not in $\{\pm 2, \pm 1\}^3$. Similarly, we can check that all half-sums of any two distinct integers in the same block of $\mathfrak{D}$ can be represented by $\mathcal{X}$ and that the coefficient set is disjoint from $\{\pm 2, \pm 1\}^3$.

The minimum and maximum values of $\mathfrak{D}$ are -62 and 62 respectively. To ensure that these two sets $[-62 : 0]$ and $[1 : 62]$ do not overlap in $\mathbb{Z}_v$, the value of v must satisfy $v \geq 2 \times 62 + 1 = 125$, since 0 is also in $\mathbb{Z}_v$. Next, let we verify that $(\mathbb{Z}_{125}, \mathfrak{D})$ is an NHSDP where $\mathfrak{D}$ has listed in (10). It is clearly that there are no intersection for any two blocks $\mathcal{D}_{\mathbf{a}}$ since each integer is uniquely represented by $\mathcal{X}$, i.e., the first condition of Definition 2 holds. Next, the second condition of Definition 2 is guaranteed by the fact that the coefficient set generated by all the integers in $\mathfrak{D}$ is disjoint from the coefficient set generated by all half-sums for any two integers in the same block. So, $(\mathbb{Z}_{125}, \mathfrak{D})$ is $(125, 8, 8)$ NHSDP.

By Theorem 1 we have a $(125, 125, 61, 1000)$ PDA which generates a 125-division $(125, M, N)$ coded caching scheme with $M/N = \frac{61}{125}$, coded caching gain $g = 8$, and the transmission load $b = 8$. Now let us consider the existing schemes with $K = 125$.

- The WCWL scheme: When $K = 125$ and $t = 61$ in [14] we have a WCWL scheme with $K = 125$, the subpacketization $F = 125$, and the transmission load $\frac{K-t}{2\lfloor \frac{K}{K-t+1} \rfloor} = \frac{64}{3} \approx 21.333 > 8$. Clearly our scheme has a lower transmission load while maintaining a lower memory ratio and the same subpacketization;
- The CWZW scheme: In the case of $q = 2$ and $m = 7$, the scheme in [12] includes the schemes in [9], [21], [22]. So we only need consider the CWZW scheme. First we can obtain a CWZW scheme with $K = 16$, the memory ratio $M/N = 0.5 > \frac{61}{125}$, the subpacketization $F' = 2^7 = 128 > 125$, and the transmission load 1. Using the grouping method in Lemma 2, we have a grouping CWZW scheme with $K = 125$, $M/N = 0.5 > \frac{61}{125}$, the subpacketization $F_1 = \frac{16}{\gcd(125,16)} \cdot F' = 2048 > 125$, and the transmission load $\frac{125}{16} = 7.8125$. We can see that our scheme has a lower memory ratio and a smaller subpacketization than the scheme in [12], while albeit with a slight increase in the transmission load;
- The MN scheme: An exhaustive computer search shows that we can obtain a $(125, M, N)$ grouping MN coded caching scheme with $M/N = 0.5$, an appropriate subpacketization $\binom{10}{5} = 252$, and the transmission load $R = \frac{5}{6} \times \frac{125}{10} = \frac{125}{12} > 10 > 8$ by using a $(10, M, N)$ MN scheme based on the grouping method in Lemma 2. It is evident that all the memory ratio, the subpacketization, and the transmission load are larger than our proposed scheme.

□

From Construction 2, for any n and m_1, m_2, ..., m_n are positive integers, suppose the vector $\mathbf{a} = (a_1, a_2, \ldots, a_n) = (m_1, m_2, \ldots, m_n) \in \mathcal{A}$. The corresponding block $\mathcal{D}_{\mathbf{a}}$ is given by

$$\mathcal{D}_{\mathbf{a}} = \{\alpha_1 f(1) + \alpha_2 f(2) + \cdots + \alpha_n f(n) \mid \alpha_i \in \{-1, 1\}, i \in [n]\},$$

which contains the minimum value $-f(1) - f(2) - \cdots - f(n)$ and the maximum value $f(1) + f(2) + \ldots + f(n)$. To ensure that $[-f(1) - f(2) - \cdots - f(n) : 0]$ and $[1 : f(1) + f(2) + \ldots + f(n)]$ can not overlap in $\mathbb{Z}_v$, the value of v must satisfy

$$v \geq 2(f(1) + f(2) + \ldots + f(n)) + 1.$$

For the ease of further notations, we define that

$$\begin{aligned}
\phi(m_1, m_2, \ldots, m_n) :=& \sum_{i=1}^{n} f(i) = \sum_{i=1}^{n-1} f(i) + m_n \left(\sum_{j=1}^{n-1} 2f(j) + 1 \right) = (1 + 2m_n) \sum_{i=1}^{n-1} f(i) + m_n \\
=& (1 + 2m_n)(1 + 2m_{n-1}) \sum_{i=1}^{n-2} f(i) + m_{n-1}(1 + 2m_n) + m_n \\
=& \prod_{i=n-2}^{n} (1 + 2m_i) \sum_{i=1}^{n-3} f(i) + m_{n-2} \prod_{i=n-1}^{n} (1 + 2m_i) + m_{n-1}(1 + 2m_n) + m_n \\
=& \sum_{i=1}^{n-1} \left(m_i \prod_{j=i+1}^{n} (1 + 2m_i) \right) + m_n. \qquad (11)
\end{aligned}$$

From (11) when $v \geq 2\phi(m_1, m_2, \ldots, m_n) + 1$ we have the following lemma, whose proof is given in Appendix B.

Lemma 3. For any n and m_1, m_2, ..., m_n are positive integers and for any odd positive integer $v \geq 2\phi(m_1, m_2, \ldots, m_n) + 1$ defined in (11), the pair $(\mathbb{Z}_v, \mathfrak{D})$ generated in Construction 2 is a $(v, 2^n, \prod_{i=1}^{n} m_i)$ NHSDP. □

By using the above NHSDP construction, we can obtain the following PDA.

Theorem 2. For any positive integer n and n positive integers m_1, m_2, ..., m_n, given a $(v, 2^n, \prod_{i=1}^n m_i)$ NHSDP, we can obtain a v-division (v, M, N) coded caching scheme with memory ratio $M/N = 1 - \frac{2^n \prod_{i=1}^n m_i}{v}$, coded caching gain $g = 2^n$, and transmission load $R = \prod_{i=1}^n m_i$, under the constraint of $v \geq 2\phi(m_1, m_2, \ldots, m_n) + 1$. □

Next given the parameters v and n (recall that $K = v$ and $g = 2^n$), we consider the selection of $m_1, \ldots, m_n$, such that the memory ratio is minimum. In other words, fixing the coded caching gain, we aim to search the coded caching scheme requiring the minimum memory ratio. Considering the constraint $v \geq 2\phi(m_1, m_2, \ldots, m_n) + 1$ into the optimization problem for the selection of $m_1, m_2, \ldots, m_n$, we have

$$\begin{aligned} \text{Problem 1.} \quad & \textbf{Maximize} \text{ function } f = \prod_{i=1}^{n} m_i \\ & \textbf{Constrains: } m_1, \ldots, m_n \in \mathbb{Z}^+, \\ & \sum_{i=1}^{n-1} \left(m_i \prod_{j=i+1}^{n} (1 + 2m_i) \right) + m_n \leq \frac{v-1}{2}. \end{aligned} \tag{12}$$

Problem 1 is an integer programming problem, which is NP-hard. We first relax the constraint $m_1, m_2, \ldots, m_n \in \mathbb{Z}^+$ to $m_1, m_2, \ldots, m_n \in \mathbb{R}^+$, to simplify the problem to a convex optimization problem, which is then solved by the Lagrange Multiplier Method. Finally we take the floor operation to the solution. For the ease of notation, we define that $q := K^{1/n}$. The following theorem introduces the sub-optimal solution by the above optimization strategy, whose proof is given in Appendix C.

Theorem 3. A sub-optimal solution (with closed-form) to Problem 1 is

$$m_1 = m_2 = \cdots = m_n = \left\lfloor \frac{v^{1/n} - 1}{2} \right\rfloor = \left\lfloor \frac{q-1}{2} \right\rfloor. \tag{13}$$

Under the selection in (13), the resulting PDA has the parameters

$$K = q^n, \quad F = q^n, \quad Z = q^n - \left(2 \left\lfloor \frac{q-1}{2} \right\rfloor \right)^n, \quad S = \left\lfloor \frac{q-1}{2} \right\rfloor^n q^n,$$

which can realize a coded caching scheme with the memory ratio $\frac{M}{N} = 1 - \left(\frac{2^n \lfloor \frac{q-1}{2} \rfloor^n}{q^n} \right)$ and the transmission load $R = \lfloor \frac{q-1}{2} \rfloor^n$. □

By Theorem 3, there are $\lfloor \log_3 K \rfloor$ memory-rate ratio points for each user number K. In addition, when q equals to a positive odd integer such that $q \geq 3$, the solution in (13) is an optimal solution to Problem 1. Then by Theorem 3 we have the following result.

Remark 2 (Optimal solution of Problem 1)**.** When q in Theorem 3 is an odd integer, we have a $(K = q^n, F = q^n, Z = q^n - (q-1)^n, S = (\frac{q-1}{2})^n q^n)$ PDA which realizes a (K, M, N) coded cachign scheme with the memory ratio $M/N = 1 - (\frac{q-1}{q})^n$, subpacketization $F = q^n$, and transmission load $R = (\frac{q-1}{2})^n$.

Given a PDA, the authors in [11] pointed out that its corresponding conjugate PDAs can be obtained in the following.

Lemma 4. *[11]* Gvien a (K, F, Z, S) PDA for some positive integers K, F, Z and S with $0 < Z < F$, there exists a $(K, S, S - (F - Z), F)$ PDA. □

By Theorem 3 and Lemma 4, the following new PDA can be obtained.

Corollary 1 (Conjugate PDA of Theorem 3)**.** The conjugate PDA corresponding to the PDA mentioned in Theorem 3 is a $(q^n, \lfloor \frac{q-1}{2} \rfloor^n q^n, \lfloor \frac{q-1}{2} \rfloor^n (q^n - 2^n), q^n)$ PDA which can realize a coded caching scheme with memory ratio $\frac{M}{N} = 1 - (\frac{2}{q})^n$ and the transmission load $R = \lfloor \frac{q-1}{2} \rfloor^{-n}$.

In particular, if q equals to some odd integer such that $q \geq 3$, we have PDAs in above Corollary 1 can be written as the following $(q^n, (\frac{q-1}{2})^n q^n, (\frac{q-1}{2})^n q^n - (q-1)^n, q^n)$ PDA.

V. Performance analysis

In this section, we will present theoretical and numerical comparisons with the existing schemes respectively to show the performance of our new scheme in Theorem 3.

A. Theoretical comparisons

Since the schemes in [16], [20], [24], [26], [27] have the special parameters of the user number and the memory ratio we only need to compare our scheme with the schemes in [2], [9], [12], [14], [19], [22], [28] respectively.

1) Comparison with the MN scheme in [2]: When $t = q^n - 2^n\lfloor\frac{q-1}{2}\rfloor^n$, we can obtain the MN scheme with the memory ratio $\frac{M}{N} = 1 - \frac{2^n\lfloor\frac{q-1}{2}\rfloor^n}{q^n}$, the subpacketization $F_{\text{MN}} = \binom{K}{t} = \binom{q^n}{q^n - 2^n\lfloor\frac{q-1}{2}\rfloor^n} = \binom{q^n}{2^n\lfloor\frac{q-1}{2}\rfloor^n}$, and the transmission load $R_{\text{MN}} = \frac{2^n\lfloor\frac{q-1}{2}\rfloor^n}{q^n - 2^n\lfloor\frac{q-1}{2}\rfloor^n + 1}$. By Theorem 3, we can obtain our scheme with the same memory ratio where $K = F = q^n$ and the transmission load $R = \lfloor\frac{q-1}{2}\rfloor^n$. Then the following results can be obtained,

$$\frac{F_{\text{MN}}}{F} = \frac{\binom{q^n}{2^n\lfloor\frac{q-1}{2}\rfloor^n}}{q^n} \approx \frac{q^{n2^n\lfloor\frac{q-1}{2}\rfloor^n}}{q^n} > \frac{q^{n(q-3)^n}}{q^n} = q^{n(q-3)^n - n} = K^{(q-3)^n - 1},$$

$$\frac{R_{\text{MN}}}{R} = \frac{\frac{2^n\lfloor\frac{q-1}{2}\rfloor^n}{q^n - 2^n\lfloor\frac{q-1}{2}\rfloor^n + 1}}{\lfloor\frac{q-1}{2}\rfloor^n} = \frac{2^n}{q^n - 2^n\lfloor\frac{q-1}{2}\rfloor^n + 1} < \frac{2^n}{q^n - (q-1)^n + 1} \approx \frac{1}{K} \cdot \frac{2^n}{1 - (\frac{q-1}{q})^n}.$$

Compared to the MN scheme, the multiplicative reduction amount by our subpacketization is proportional to $K^{(q-3)^n-1}$ which grows exponentially with the user number K at a rate of $(q-3)^n$, while the increase amount of our transmission is only

$$\frac{q^n - 2^n\lfloor\frac{q-1}{2}\rfloor^n + 1}{2^n} \approx \frac{q^n - (q-1)^n}{2^n},$$

which decreases exponentially with 2 at a rate of n.

2) Comparison with the WCLC scheme in [19]: The authors in [19] showed that the scheme includes the partition schemes in [9], the hypergraph schemes in [22], and the OA schemes in [12] as the special cases. So first we only need to compare with the scheme in [19]. When $z > 1$, the user number in the WCLC scheme is too complex, so we have to use some specific parameters for the comparison. When $m = z = n$, $k = q$, and $t = q - 2\lfloor\frac{q-1}{2}\rfloor$, we have the WCLC scheme in [19], where the number of users $K = q^n$ with the memory ratio, the subpacketization, and the transmission load are respectively

$$\frac{M}{N} = 1 - \frac{2^n\lfloor\frac{q-1}{2}\rfloor^n}{q^n}, \quad F_{\text{WCLC}} = \frac{q^{n-1}(q-1)^n}{2^n\lfloor\frac{q-1}{2}\rfloor^n}, \quad R_{\text{WCLC}} = \frac{2^n\left\lfloor\frac{q-1}{2}\right\rfloor^n}{\lfloor\frac{q-1}{2\lfloor\frac{q-1}{2}\rfloor}\rfloor^n}.$$

By Theorem 3, we can obtain a coded caching scheme with the same memory ratio and the same number of users, the subpacketization $F = q^n$, and the transmission load $R = \lfloor\frac{q-1}{2}\rfloor^n$. Then the following result can be obtained,

$$\frac{F_{\text{WCLC}}}{F} = \frac{\frac{q^{n-1}(q-1)^n}{2^n\lfloor\frac{q-1}{2}\rfloor^n}}{q^n} = \frac{(q-1)^n}{q \cdot 2^n\lfloor\frac{q-1}{2}\rfloor^n} \approx \frac{1}{q}, \quad \frac{R_{\text{WCLC}}}{R} = \frac{\frac{2^n\lfloor\frac{q-1}{2}\rfloor^n}{\lfloor\frac{q-1}{2\lfloor\frac{q-1}{2}\rfloor}\rfloor^n}}{\left\lfloor\frac{q-1}{2}\right\rfloor^n} = \frac{2^n}{\lfloor\frac{q-1}{2\lfloor\frac{q-1}{2}\rfloor}\rfloor^n} < \frac{4^n\lfloor\frac{q-1}{2}\rfloor^n}{(q-1)^n} \approx 2^n.$$

Compared to the WCLC scheme, our subpacketization increases by a factor of q times while reducing the transmission load by a factor of 2^n.

3) Comparison with the WCWL scheme in [14]: Let $K = q^n$ and $t = q^n - 2^n\lfloor\frac{q-1}{2}\rfloor^n$ in [14], i.e., the 10th row of Table I. We compare the WCWL scheme with the proposed scheme in the following cases:

- If $(K-t+1)|K$ or $K - t = 1$, we have

$$F_{\text{WCWL}} = K = q^n, \quad R_{\text{WCWL}} = \frac{(K-t)(K-t+1)}{2K} = \frac{2^n\lfloor\frac{q-1}{2}\rfloor^n(2^n\lfloor\frac{q-1}{2}\rfloor^n + 1)}{2q^n}.$$

  By Theorem 3, we have the following result.

$$\frac{F_{\text{WCWL}}}{F} = \frac{q^n}{q^n} = 1, \quad \frac{R_{\text{WCWL}}}{R} = \frac{\frac{2^n\lfloor\frac{q-1}{2}\rfloor^n(2^n\lfloor\frac{q-1}{2}\rfloor^n+1)}{2q^n}}{\lfloor\frac{q-1}{2}\rfloor^n} = \frac{2^n(2^n\lfloor\frac{q-1}{2}\rfloor^n + 1)}{2q^n} > \frac{2^n(q-3)^n}{2q^n} = \frac{1}{2}\left(2 - \frac{6}{q}\right)^n.$$

- If $\langle K\rangle_{K-t+1} = K - t$, we can get the WCWL scheme with

$$F_{\text{WCWL}} = \left(2\lfloor\frac{K}{K-t+1}\rfloor + 1\right)K, \quad R_{\text{WCWL}} = \frac{K-t}{2\lfloor\frac{K}{K-t+1}\rfloor + 1} = \frac{2^n\lfloor\frac{q-1}{2}\rfloor^n}{2\left\lfloor\frac{q^n}{2^n\lfloor\frac{q-1}{2}\rfloor^n+1}\right\rfloor + 1}.$$

By Theorem 3, we can obtain the following.

$$\frac{F_{\text{WCWL}}}{F}=\frac{(2\lfloor\frac{K}{K-t+1}\rfloor+1)K}{q^n}=2\left\lfloor\frac{K}{K-t+1}\right\rfloor+1,$$

$$\frac{R_{\text{WCWL}}}{R}=\frac{\frac{2^n\lfloor\frac{q-1}{2}\rfloor^n}{2\left\lfloor\frac{q^n}{2^n\lfloor\frac{q-1}{2}\rfloor^n+1}\right\rfloor+1}}{\lfloor\frac{q-1}{2}\rfloor^n}=\frac{2^n}{2\left\lfloor\frac{q^n}{2^n\lfloor\frac{q-1}{2}\rfloor^n+1}\right\rfloor+1}>\frac{2^n(q-3)^n}{2q^n+(q-1)^n+1}>\frac{2^n(q-3)^n}{3q^n}=\frac{1}{3}\cdot\left(2-\frac{6}{q}\right)^n.$$

- Otherwise, the WCWL scheme can be obtain as follow.

$$F_{\text{WCWL}}=2\left\lfloor\frac{K}{K-t+1}\right\rfloor K,\quad R_{\text{WCWL}}=\frac{K-t}{2\lfloor\frac{K}{K-t+1}\rfloor}=\frac{2^n\lfloor\frac{q-1}{2}\rfloor^n}{2\left\lfloor\frac{q^n}{2^n\lfloor\frac{q-1}{2}\rfloor^n+1}\right\rfloor}.$$

By Theorem 3, we have

$$\frac{F_{\text{WCWL}}}{F}=\frac{2\left\lfloor\frac{K}{K-t+1}\right\rfloor K}{q^n}=2\left\lfloor\frac{K}{K-t+1}\right\rfloor,$$

$$\frac{R_{\text{WCWL}}}{R}=\frac{\frac{2^n\lfloor\frac{q-1}{2}\rfloor^n}{2\left\lfloor\frac{q^n}{2^n\lfloor\frac{q-1}{2}\rfloor^n+1}\right\rfloor}}{\lfloor\frac{q-1}{2}\rfloor^n}=\frac{2^n}{2\left\lfloor\frac{q^n}{2^n\lfloor\frac{q-1}{2}\rfloor^n+1}\right\rfloor}>\frac{2^n(q-3)^n}{2q^n}=\frac{1}{2}\cdot\left(2-\frac{6}{q}\right)^n.$$

As a result, compared to the WCWL scheme in [14], our scheme either obtaining the same subpacketization or the reduction amount of our subpacketization is larger than 2 times. The multiplicative reduction amount by our transmission is at least $\frac{1}{3}\cdot(2-\frac{6}{q})^n$. When $q>6$ and $n\geq\frac{\ln 3}{\ln(2-\frac{6}{q})}$, the formula $\frac{1}{3}\cdot(2-\frac{6}{q})^n\geq 1$ always holds.

4) Comparison with the XXGL scheme in [28]: Let $K=q^n$ in [28], i.e., the 11th row of Table I. We have the XXGL scheme with memory ratio $M/N=1-\frac{2}{q^n}\gg 1-(\frac{2}{q})^n$, the subpacketization $F_{\text{XXGL}}=q^n$, and the transmission load $R_{\text{XXGL}}=\frac{q^n-1}{q^n}$. The scheme in Corollary 1 have $F=\lfloor\frac{q-1}{2}\rfloor^n q^n$ and $R=\lfloor\frac{q-1}{2}\rfloor^{-n}$. So we have

$$\frac{F_{\text{XXGL}}}{F}=\frac{q^n}{\lfloor\frac{q-1}{2}\rfloor^n q^n}=\left\lfloor\frac{q-1}{2}\right\rfloor^{-n},\qquad\frac{R_{\text{XXGL}}}{R}=\frac{\frac{q^n-1}{q^n}}{\lfloor\frac{q-1}{2}\rfloor^{-n}}\approx\left\lfloor\frac{q-1}{2}\right\rfloor^n.$$

Compared to the XXGL scheme in [28], even though our memory ratio is much smaller than that of the XXGL scheme, the reduction amount of our transmission load is about $\lfloor\frac{q-1}{2}\rfloor^n$ times while our subpacketization increases $\lfloor\frac{q-1}{2}\rfloor^n$ times which is much less than K. So our scheme has the significant advantages on transmission load compared to the XXGL scheme.

B. Numerical comparisons

In this subsection, we will present numerical comparisons between our scheme introduced in Theorem 3 and the existing schemes that employ linear subpacketization [14], [16], [20], [27], [28], polynomial subpacketization [24], [26], and exponential subpacketization [2], [19]. These schemes are summarized in Table I. It is worth noting that the ASK scheme [27] can be included as a special case of our scheme in Corollary 2 which will be introduced in Subsection VI-A. In addition, the XXGL scheme [28] has large memory ratio which approximates 1 so in this subsection we do not need to consider the XXGL scheme.

1) Comparison with the linear subpacketization schemes in [14], [16], [20]: In Table II, we present some numerical comparisons with the schemes in [16], [20]. Compared to the ZCW scheme, our scheme in Theorem 3 has more users, a close memory ratio, a similar subpacketization, and a smaller load. Compared to the AST scheme, our scheme has a smaller memory ratio, a lower transmission load, and a smaller subpacketization while having more users.

Now, let us compare our scheme in Theorem 3 with the WCWL scheme in [14]. When $K=85$, we show our scheme (represented by the red line) and the WCWL scheme (represented by the green line) in Figure 2 and Figure 3. We can see that our scheme has a smaller or equal subpacketization, and a lower transmission load for the memory ratio is less than 0.9. When the memory ratio is close to 1, our scheme has a slightly higher transmission load while having a smaller or equal subpacketization.

Finally, we would like to point out that the advantages of our scheme in terms of the subpacketization and transmission load becomes more obvious as the number of users K increases. For instance, when $K=729$, we show our scheme and the WCWL scheme in Figure 4 and Figure 5. We can observe that our scheme has a smaller or equal subpacketization and a smaller transmission load.

TABLE II: The numerical comparison between the scheme in Theorem 3 and the schemes in [16], [20]

K	M/N	Scheme	Parameters	Load	Subpacketization
32	0.6875	ZCW scheme in [16]	$(m,w)=(5,2)$	1.25	32
33	0.7576	Scheme in Theorem 3	$(v,n)=(33,3)$	1	33
128	0.836	ZCW scheme in [16]	$(m,w)=(7,2)$	2.625	128
129	0.876	Scheme in Theorem 3	$(v,n)=(129,4)$	1	129
256	0.8906	ZCW scheme in [16]	$(m,w)=(8,2)$	2.41	256
257	0.8755	Scheme in Theorem 3	$(v,n)=(257,5)$	1	257
512	0.9297	ZCW scheme in [16]	$(m,w)=(9,2)$	2.04	512
513	0.9376	Scheme in Theorem 3	$(v,n)=(513,5)$	1	513
52	0.28846	AST scheme in [20]	$(r,k)=(2,13)$	9.25	52
49	0.26531	Scheme in Theorem 3	$(v,n)=(49,2)$	9	49
1332	0.2515	AST scheme in [20]	$(r,k)=(2,333)$	249.25	1332
1331	0.2498	Scheme in Theorem 3	$(v,n)=(1331,3)$	125	1331
2192	0.2509	AST scheme in [20]	$(r,k)=(2,548)$	410.5	2192
2199	0.2142	Scheme in Theorem 3	$(v,n)=(2199,3)$	216	2199
2400	0.50125	AST scheme in [20]	$(r,k)=(3,300)$	149.625	2400
2401	0.460	Scheme in Theorem 3	$(v,n)=(2401,4)$	81	2401

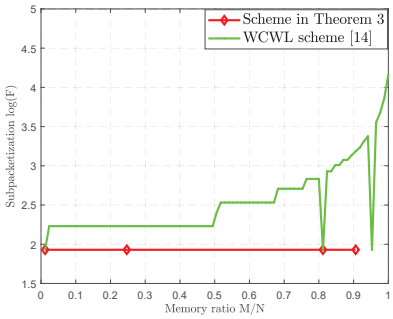

Fig. 2: Memory ratio-subpacketization tradeoff for $K = 85$

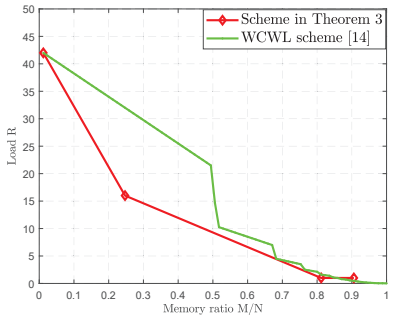

Fig. 3: Memory ratio-load tradeoff for $K = 85$

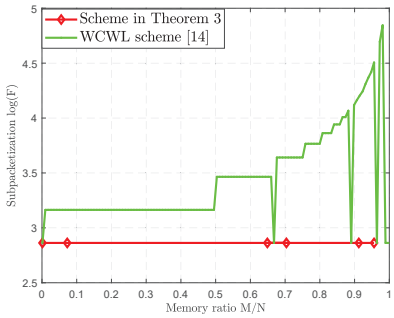

Fig. 4: Memory ratio-subpacketization tradeoff for $K = 729$

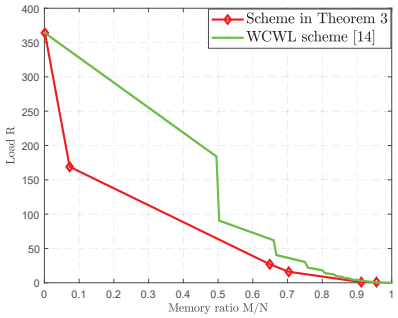

Fig. 5: Memory ratio-load tradeoff for $K = 729$

2) Comparison with the polynomial subpacketization schemes in [24], [26]: Since the user numbers of the schemes in [24], [26] are highly specialized integers, including combination numbers, powers, products of combination numbers, and powers, it is difficult for us to directly plot a graph for given values of K. Instead, we compare our scheme with the CKSM scheme and the YTCC scheme by choosing some specific user numbers and memory ratios listed in Table III. We can see that compared to the YTCC scheme, our scheme has a similar memory ratio, a slightly larger transmission load, and more users, while having a much smaller subpacketization. Compared to the CKSM scheme, our scheme has a lower transmission load and a much smaller subpacketization while having a close number of users and a close memory ratio.

3) Comparison with the exponential subpacketization schemes in [2], [19]: Let us first directly compare our scheme in Theorem 3 with the MN scheme and the WCLC scheme, each of which has an exponential subpacketization level with the number of users. When $K = 85$, we have our scheme (represented by the red line), the MN scheme (represented by the blue line), and the WCLC scheme (represented by the green line) listed in Figure 6 and Figure 7 by Table I and Theorem 3.

Compared to the MN scheme, we can observe that our subpacketization (the red line) is much smaller than the MN scheme (the blue line), while our transmission load is little larger than the MN scheme when $K = 85$. Compared to the WCLC scheme, our scheme (the red line) has a much smaller subpacketization than the WCLC scheme (the green line) while our transmission

TABLE III: The numerical comparison between the schemes in [24], [26] and the scheme in Theorem 3

K	M/N	Scheme	Parameters	Load	Subpacketization
22	0.364	YTCC scheme in [24]	$(H,a,b,r)=(22,1,8,0)$	1.556	319770
25	0.36	Scheme in Theorem 3	$(v,n)=(25,2)$	4	25
78	0.81	YTCC scheme in [24]	$(H,a,b,r)=(13,2,7,0)$	0.41667	1716
81	0.802	Scheme in Theorem 3	$(v,n)=(81,4)$	1	81
105	0.486	YTCC scheme in [24]	$(H,a,b,r)=(15,2,9,1)$	6	5005
127	0.496	CKSM scheme in [26]	$(q,k,m,t)=(2,7,6,1)$	9.143	$3.56E+09$
125	0.488	Scheme in Theorem 3	$(v,n)=(125,3)$	8	125
325	0.354	YTCC scheme in [24]	$(H,a,b,r)=(26,2,5,0)$	10	65780
343	0.370	Scheme in Theorem 3	$(v,n)=(343,3)$	27	343
231	0.805	YTCC scheme in [24]	$(H,a,b,r)=(22,2,12,0)$	0.4945055	646646
255	0.8784	CKSM scheme in [26]	$(q,k,m,t)=(2,8,4,1)$	2.0667	97155
243	0.8683	Scheme in Theorem 3	$(v,n)=(243,5)$	1	243
31	0.7742	CKSM scheme in [26]	$(q,k,m,t)=(2,5,2,1)$	1	155
27	0.7037	Scheme in Theorem 3	$(v,n)=(27,3)$	1	27
85	0.247	CKSM scheme in [26]	$(q,k,m,t)=(4,4,3,1)$	16	95200
81	0.210	Scheme in Theorem 3	$(v,n)=(81,2)$	16	81
156	0.19872	CKSM scheme in [26]	$(q,k,m,t)=(5,4,3,1)$	31.25	604500
121	0.1736	Scheme in Theorem 3	$(v,n)=(121,2)$	25	121
156	0.1987	CKSM scheme in [26]	$(q,k,m,t)=(5,4,3,1)$	31.25	604500
169	0.1479	Scheme in Theorem 3	$(v,n)=(169,2)$	36	169
255	0.12157	CKSM scheme in [26]	$(q,k,m,t)=(2,8,5,1)$	37.333	$8.10E+09$
225	0.1289	Scheme in Theorem 3	$(v,n)=(225,2)$	49	225
364	0.332	CKSM scheme in [26]	$(q,k,m,t)=(3,6,5,1)$	40.5	$4.51E+10$
343	0.370	Scheme in Theorem 3	$(v,n)=(343,3)$	27	343

load is close to or equal to that of the WCLC scheme when $K=85$.

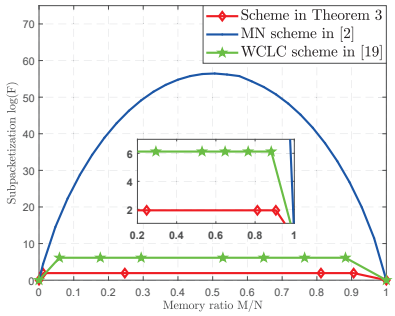

Fig. 6: Memory ratio-subpacketization tradeoff for $K=85$

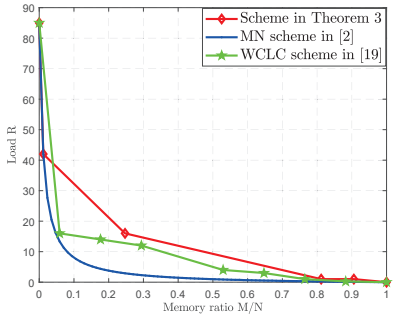

Fig. 7: Memory ratio-load tradeoff for $K=85$

Using the grouping method in Lemma 2 on the schemes in [2], [19], through an exhaustive search on K_1 given $K=85$, we find that we cannot obtain a scheme with both a lower load and a smaller subpacketization.

Finally we should point out that the advantages on the subpacketization and transmission of our scheme are more obvious as K increases. For instance, when $K=729$ we compare our scheme with the MN scheme and the WCLC scheme in Figure 8 and Figure 9. Note that the ratio of the subpacketization between our scheme and WCLC scheme significantly reduces (see Figure 6 and Figure 8), and our transmission load is more close to that of the WCLC (see Figure 7 and Figure 9).

VI. Relationship between NHSDP and the classic combinatorial Designs

In this section, we will show that there is a close relationship between NHSDP and several classical combinatorial designs, such as cyclic difference packing [31], non-three-term arithmetic progressions [32], and perfect hash families [40]. As a result, some new NHSDP can be obtained by the existing cyclic difference packings and some new classes of non-three-term arithmetic progression sets and perfect hash families are obtained by Theorem 3.

A. The cyclic difference packing

Definition 3 (CDP [31])**.** A (v,k) cyclic difference packing (CDP) is a pair $(\mathbb{Z}_v,\mathcal{D})$ where the subset $\mathcal{D}=\{d_1,d_2,\ldots,d_k\}$ of $\mathbb{Z}_v$ satisfies that each non-zero integer has at most one representation as a difference d_i-d_j. If each non-zero integer has exactly one representation as a difference d_i-d_j, $(\mathbb{Z}_v,\mathcal{D})$ is called (v,k) difference set (DS). □

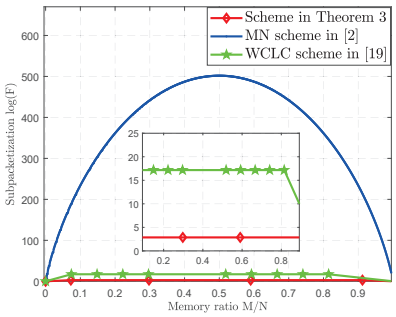

Fig. 8: Memory ratio-subpacketization tradeoff for $K = 729$

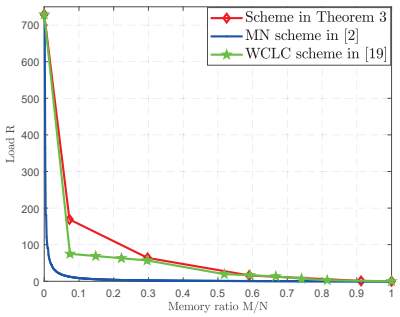

Fig. 9: Memory ratio-load tradeoff for $K = 729$

For instance when $v = 7$ and $k = 3$, we can obtain a pair $(\mathbb{Z}_7, \mathcal{D} = \{0, 1, 3\})$. In addition we have the following relationships.

$$1 = 1 - 0,\ 2 = 3 - 1,\ 3 = 3 - 0,\ 4 = 0 - 3,\ 5 = 0 - 2,\ 6 = 0 - 1,$$

i.e., each non-zero integer has exactly one representation as a difference $d_i - d_j$ for each $1 \le i \ne j \le 3$. So $(\mathbb{Z}_7, \mathcal{D})$ is a $(7, 3)$ CDP and is also a $(7, 3)$ DS. In addition we can check that $(\mathbb{Z}_7, \mathcal{D})$ is also a $(7, 3, 1)$ NHSDP since $\{\frac{0+1}{2} = 4, \frac{0+3}{2} = 5, \frac{1+3}{2} = 2\} \cap \{0, 1, 3\} = \emptyset$. In general, the following statement always holds.

Lemma 5. Any (v, k) CDP is a $(v, g = k, 1)$ NHSDP. □

Proof. Assume that $(\mathbb{Z}_v, \mathcal{D} = \{d_1, d_2, \ldots, d_k\})$ is a (v, k) CDP. In order to show that it is also an NHSDP, we only need to consider the half-sum of any two different elements in $\mathcal{D}$. Suppose that there are three different elements, say x, y and z, of $\mathcal{D}$ that satisfy $2z = x + y$. Then we have $a = z - y = x - z$ which is impossible by Definition 3, i.e., each non-zero integer has at most one representation as a difference of two different elements from $\mathcal{D}$. Then our proof is complete. □

It is well known that for any prime power q, there always exists a $(q^2 + q + 1, q + 1)$ DS. By Lemma 5 we have a $(q^2 + q + 1, q + 1, 1)$ NHSDP. Then by Theorem 1 and Lemma 1 the following result can be obtained.

Corollary 2. For any prime power q, there exists a $(q^2+q+1, q^2+q+1, q+1, q^2+q+1)$ PDA which realizes a (q^2+q+1)-division $(K = q^2 + q + 1, M, N)$ coded caching scheme with $M/N = \frac{q^2}{q^2+q+1}$ and transmission load $R = 1$. □

It is worth noting that the scheme in Corollary 2 has the same system parameters K, M and N and the same performance, i.e., the same subpacketization and transmission load, as the ASK scheme in [27]. The reason is that the scheme in [27] is generated by a symmetric balanced incomplete block design (SBIBD) which is can be constructed by a DS [41].

B. Expanding the Concept of NTAP

NHSDP includes the well-known non-three-term arithmetic progressions (NTAP) [32] as a special case. A subset $\mathcal{S} \subseteq \mathbb{Z}_v$ is called an NTAP set if any three different elements, say x, y and z, of $\mathcal{S}$ do not have the property $2z = x + y$. If $\mathcal{S}$ is an NTAP set, $(\mathbb{Z}_v, \mathcal{S})$ is an NHSDP with one block. So in Theorem 3, when $q = 3$, we have $b = 1$. In this case, (25) can be written as follows.

$$\mathcal{D} = \{\alpha_1 + 3\alpha_2 + \cdots + 3^{n-1}\alpha_n \mid \alpha_i \in \{-1, 1\}, i \in [n]\}.$$

By Theorem 3 we know that $\mathcal{D}$ is an NTAP subset of $\mathbb{Z}_v$. Then the following result can be obtained.

Lemma 6 (NTAP set)**.** For any positive integer n, there exists an NTAP with size of $\rho_1 = 2^n$ over $\mathbb{Z}_{3^n}$.

For any given positive integer v define the maximum size of an NTAP subset of $\mathbb{Z}_v$ by $\rho(v)$. The authors in [36] showed that $\rho(v) \ge \rho_2 = v \cdot 2^{-(2\sqrt{\log_2(24/7)}+o(1))\sqrt{\log_2 v}}$. As far as we know, this is the best lower bound on $\rho(v)$ up to now. When $v = 3^n$, we have

$$\frac{\rho_2}{\rho_1} = \frac{v \cdot 2^{-(2\sqrt{\log_2(24/7)}+o(1))\sqrt{\log_2 v}}}{2^n} = \frac{3^n}{2^{(2\sqrt{\log_2(24/7)}+o(1))\sqrt{n \log_2 3}+n}}.$$

Taking the logarithm of the above ratio, we have

$$\begin{aligned}\ln\left(\frac{\rho_2}{\rho_1}\right) =& n\ln 3-((2\sqrt{\log_2(24/7)}+o(1))\sqrt{n\log_2 3}+n)\ln 2\\ =& 1.0986n-0.6931(2.6665+o(1))\cdot 1.5850n^{1/2}+n)\\ =& 1.0986n-0.6931n-2.9293n^{1/2}-o(1)\cdot 1.5850n^{1/2}\\ =& 0.4055n-2.9293n^{1/2}-o(1)\cdot 1.5850n^{1/2}.\end{aligned}$$

By the above formula, we have $2^n \geq \frac{v\cdot 2^{-(2\sqrt{\log_2(24/7)}+o(1))\sqrt{\log_2 v}}}{2^n}$ when $n \leq \lfloor\left(\frac{2.9293}{0.4055}\right)^2\rfloor = 52$,. That is when $n \leq 52$, $\rho_1 > \rho_2$ always holds; in other words, by our construction, we propose a tighter lower bound on $\rho(v)$, i.e., $\rho(v) \geq \rho_1$, than the existing one.

C. Perfect hash family

Perfect hash family is a useful combinatorial structure which is generalization of many well-studied objects in combinatorics, cryptography, and coding theory and has many interesting applications, for example, in secure frame-proof codes of fingerprinting problems [40], threshold cryptography [42], [43], covering arrays used in software testing problems [44], broadcast encryptions [45], etc.

Let φ be a function from a set $\mathcal{B}$ to a set $\mathcal{C}$. We say that f separates a subset $\mathcal{T} \subseteq \mathcal{B}$ if φ is injective when restricted to $\mathcal{T}$. Let m, q, t be integers such that $m \geq q \geq t \geq 2$. Suppose $|\mathcal{B}| = m$ and $|\mathcal{C}| = q$. A set $\mathcal{X}$ of functions from $\mathcal{B}$ to $\mathcal{C}$ with $|\mathcal{X}| = r$ is an $(r; m, q, t)$-perfect hash family if for all $\mathcal{T} \subseteq \mathcal{B}$ with $|\mathcal{T}| = t$, there exists at least one $\varphi \in \mathcal{X}$ such that φ separates $\mathcal{T}$. We refer to this as an $(r; m, q, t)$ PHF. An $(r; m, q, t)$ PHF can be depicted as an $r \times m$ array in which the columns are labeled by the elements of $\mathcal{B}$, the rows by the functions $\varphi_i \in \mathcal{X}$ and the (i, j)-entry of the array is the value $\varphi_i(j)$. Thus, an $(r; m, q, t)$ PHF is equivalent to an $r \times m$ array with entries from a set of q symbols such that every $r \times t$ subarray contains at least one row having distinct symbols. A perfect hash family is considered optimal if m is as large as possible given r, q and t.

The authors in [46], [47] showed that an NTAP set of size g over $\mathbb{Z}_v$, i.e., $(v, g, 1)$ NHSDP, can also be used to construct a $(3; gv, v, 3)$ PHF. Using the $(q^2 + q + 1, q + 1, 1)$ NHSDP generated by a $(q^2 + q + 1, q + 1)$ DS for any prime power q, we have a can obtain the following new PHF.

Lemma 7. For any prime power q, there exists a $(3; m_1 = (q^2 + q + 1)(q + 1), q^2 + q + 1, 3)$ PHF. □

When $q = 3$ in Theorem 3, we have the following new PHF.

Lemma 8. For any positive integer n, there exists a $(3; m_2 = 6^n, v = 3^n, 3)$ PHF. □

By the classic generalized quadrangles, quadrics in PG$(4, p)$, and Hermitian varieties in PG$(4, p^2)$ for any prime power p, the authors in [37] constructed the following PHFs which has maximum m among the existing deterministic constructions.

Lemma 9 ([37])**.** For any prime power p, there exist a $(3; m_3 = p^2(p + 1), v_1 = p^2, 3)$ quadrics PHF and a $(3; m_4 = p^5, v_2 = p^3, 3)$ Hermitian PHF.

By Lemma 7 and Lemma 9, we have $m_1 \approx m_3$ when $p = q$. Now let us compare the values of m_2, m_3 and m_4 in Lemma 8 and Lemma 9. Let $v = p^2 = 3^n$ and $v = p^3 = 3^n$ respectively where $n \geq 2$. We have

$$\frac{m_3}{m_2} = \frac{3^n(3^{n/2}+1)}{6^n} = \left(\frac{3}{4}\right)^{n/2} + \frac{1}{2^n} < 1 \quad \text{and} \quad \frac{m_4}{m_2} = \frac{3^{5n/3}}{6^n} = \frac{3^{2n/3}}{2^n} \approx 1.04^n.$$

This implies that the PHF proposed in Lemma 8 achieves more columns than the first quadrics PHF and approaches the number of column the Hermitian PHF in [37].

VII. Conclusion

In this paper, based on the Latin square we introduced a new combinatorial structure called non-half-sum disjoint packing (NHSDP) to construct PDA where the subpacketization is linear with K. According to theoretical and numerical comparisons, the proposed scheme achieves lower load than the existing schemes with linear subpacketization; it achieves lower load in some cases than some existing schemes with polynomial subpacketization; it has close loads as some existing schemes with exponential subpacketization in some cases. Furthermore, NHSDP has a close relationship with the classical combinatorial structures such as cyclic difference packing, non-three-term arithmetic progressions and perfect hash family (PHP) and so on. By constructing NHSDPs, we can obtain some new CDPs, NTAPs and PHPs with large cardinalities or columns.

Finally, in this paper we only focus on studying the (v, g, b) NHSDP with $g = 2^n$ for any positive integer n. So it is meaningful to study the (v, g, b) NHSDP for any value of g.

APPENDIX A
PROOF OF THEOREM 1

Assume that $(\mathbb{Z}_v, \mathfrak{D})$ is a (v, g, b) NHSDP where $\mathfrak{D} = \{\mathcal{D}_1, \mathcal{D}_2, \ldots, \mathcal{D}_b\}$. According to Definition 1, let us compute the number of stars in each column first. For each column k and each each integer $x \in \mathfrak{D}$, there exactly exists a unique integer f satisfying that $k - f = x$. This implies that there are exactly bg integer entries in column k, i.e., $v - bg$ star entries in column k. So we have $Z = v - bg$.

Next, for any two different entries say p_{f_1,k_1} and p_{f_2,k_2} satisfying that $p_{f_1,k_1} = p_{f_2,k_2} = (c, i)$, from (6) we have

$$d_1 = k_1 - f_1,\ d_2 = k_2 - f_2 \in \mathcal{D}_i, \quad \text{and} \quad k_1 + f_1 = k_2 + f_2. \tag{14}$$

If $f_1 = f_2$ (or $k_1 = k_2$) we have $k_1 = k_2$ (or $f_1 = f_2$) which contradicts our hypothesis that p_{f_1,k_1} and p_{f_2,k_2} are two different entries. So each integer pair in $\mathbf{P}$ occurs in each row and each column at most once. Let us consider the case $f_1 \neq f_2$ and $k_1 \neq k_2$. First from (14), we have $k_1 = f_1 + d_1, k_2 = f_2 + d_2$ and

$$2f_1 + d_1 = 2f_2 + d_2, \quad \text{i.e.,} \quad d_1 - d_2 = 2(f_2 - f_1). \tag{15}$$

Assume that the entry p_{f_1,k_2} is not star. Then from (14) there exists a integer $i' \in [b]$ and $d_3 \in \mathbb{Z}_v$ such that $d_3 = k_2 - f_1 \in \mathcal{D}_{i'}$. This implies that $k_2 = d_3 + f_1$. Together with $k_2 = d_2 + f_2$ we have $d_3 - d_2 = f_2 - f_1$. From (15) we have $2(d_3 - d_2) = d_1 - d_2$, i.e., $2d_3 = d_1 + d_2$. This implies that $d_3 = \frac{d_1+d_2}{2} \in \mathcal{D}_{i'}$ when v is odd. This contracts the condition of NHSDP that the half-sum of any two different elements in $\mathcal{D} \in \mathfrak{D}$ dose not occur in $\mathfrak{D}$. Similarly we can also show that $p_{f_2,k_1} = *$.

Finally let us compute the number of different integer pairs in $\mathbf{P}$. For any integers $i \in [b]$, $c \in \mathbb{Z}_v$ and for each integer $d \in \mathcal{D}_i$, when v is odd, the following system of equations always has a unique solution:

$$\begin{cases} k - f = d; \\ k + f = c. \end{cases}$$

When d runs all the element of $\mathcal{D}_i$there exactly g unique solutions. This means that the integer pair (c, i) occurs in $\mathbf{P}$ exactly g times and there are exactly bv different integer pairs. Then the proof is completed.

APPENDIX B
PROOF OF LEMMA 3

By Definition 2 let us consider the first condition that the intersection of any two difference blocks in $\mathfrak{D}$ is empty. Let $\mathbf{a} = (a_1, a_2, \ldots, a_n)$, $\mathbf{a}' = (a'_1, a'_2, \ldots, a'_n)$, $\alpha = (\alpha_1, \alpha_2, \ldots, \alpha_n)$, and $\alpha' = (\alpha'_1, \alpha'_2, \ldots, \alpha'_n)$ are four vectors, where $\mathbf{a}, \mathbf{a}' \in \mathcal{A}$ and $\alpha, \alpha' \in \{1, -1\}^n$. From (9) we have two integers of $\mathfrak{D}$, i.e.,

$$x = \alpha_1 a_1 x_1 + \cdots + \alpha_n a_n x_n \in \mathcal{D}_{\mathbf{a}} \quad \text{and} \quad y = \alpha'_1 a'_1 x_1 + \cdots + \alpha'_n a'_n x_n \in \mathcal{D}_{\mathbf{a}'}. \tag{16}$$

Since $v \geq 2\phi(m_1, m_2, \ldots, m_n) + 1 = 2\sum_{i=1}^{n} f(i) + 1$, both $\sum_{i=1}^{n} a_i x_i$ and $\sum_{i=1}^{n} a'_i x_i$ are less tha $\frac{v-1}{2}$. In addition, from (8) for each $i \in [n-1]$ we have $x_{i+1} > 2\sum_{j=1}^{i} f(j)$ which implies that

$$a_{i+1} x_{i+1} > 2 \sum_{j=1}^{i} a_j x_j. \tag{17}$$

If $x = y$, we have $\alpha_n = \alpha'_n$ and $a_n = a'_n$. Otherwise, from (17) with $i = n - 1$, if $a_n \neq a'_n$ we have $x \neq y$. Furthermore, if $\alpha_n \neq \alpha'_n$, without loss of generality, we assume that $\alpha_n < 0$ and $\alpha'_n > 0$. Then we have $\frac{v-1}{2} < x < v$ and $y \leq \frac{v-1}{2}$, which implies $x \neq y$. This contradicts our hypothesis that $x = y$. So we only need to consider the case $x - \alpha_n a_n x_n = y - \alpha'_n a'_n x_n$. Similarly we can obtain $a_{n-1} = a'_{n-1}$ and $\alpha_{n-1} = \alpha'_{n-1}$ from (17) with $i = n - 2$. Using the aforementioned proof method, we can analogously obtain $a_i = a'_i$ and $\alpha_i = \alpha'_i$, which implies $\alpha_i a_i x_i = \alpha'_i a'_i x_i$ for each integer $i \in [n]$. Then $\mathbf{a} = \mathbf{a}'$ and $\alpha = \alpha'$. So each integer in $\mathfrak{D}$ appears exactly once.

Now let us check the property of non-half-sum. For any vector $\mathbf{a} = (a_1, a_2, \ldots, a_n) \in \mathcal{A}$ and any two different vectors $\alpha = (\alpha_1, \alpha_2, \ldots, \alpha_n)$, $\alpha' = (\alpha'_1, \alpha'_2, \ldots, \alpha'_n) \in \{-1, 1\}^n$, let us consider the half-sum of integers

$$x = \alpha_1 a_1 x_1 + \cdots + \alpha_n a_n x_n \quad \text{and} \quad y = \alpha'_1 a_1 x_1 + \cdots + \alpha'_n a_n x_n, \tag{18}$$

i.e.,

$$\frac{x + y}{2} = \frac{\alpha_1 + \alpha'_1}{2} \cdot a_1 x_1 + \cdots + \frac{\alpha_n + \alpha'_n}{2} \cdot a_n x_n.$$

By our hypothesis $\alpha \neq \alpha'$ we have $x \neq y$. In addition, since $\alpha_1, \alpha_2 \in \{-1, 1\}$ it follows that $\frac{\alpha_i + \alpha'_i}{2} \in \{-1, 0, 1\}$ for each $i \in [n]$. Furthermore there exists at least one integer $i' \in [n]$ such that $\frac{\alpha_i + \alpha'_i}{2} = 0$. Otherwise we have $\alpha = \alpha'$ which implies

$x = y$. This contradicts our hypothesis $x \neq y$. In the following we will show that $\frac{x+y}{2}$ does not occur in $\mathfrak{D}$. Assume that there exists two vectors $\mathbf{a}' = (a'_1, a'_2, \ldots, a'_n) \in \mathcal{A}$ and $\beta = (\beta_1, \beta_2, \ldots, \beta_n) \in \{-1, 1\}^n$ such that

$$z = a'_1\beta_1 x_1 + \cdots + a'_n\beta_n x_n = \frac{\alpha_1 + \alpha'_1}{2} \cdot a_1 x_1 + \cdots + \frac{\alpha_n + \alpha'_n}{2} \cdot a_n x_n.$$

Similar to the proof of the uniqueness of each integer in $\mathfrak{D}$ introduced above, we can get $\beta_i = \frac{\alpha_i + \alpha'_i}{2}$ and $a'_i = a_i$ for every $i \in [n]$. Then there exists at least one $\beta_{i'} = 0$ for some $i' \in [n]$. This contradicts our definition rule given in (9), that is, $\beta_{i'} \in \{-1, 1\}^n$. So the half-sum of any two different integers in $\mathfrak{D}$ does not occur in $\mathfrak{D}$. Then the proof is completed.

APPENDIX C
PROOF OF THEOREM 3

Clearly this is a convex n-th programming problem under the real number domain when m_1, m_2, ..., m_n are real variables. We can use the Lagrange Multiplier Method to find its optimal solution. Specifically by partially differentiating of the function

$$\psi(m_1, \ldots, m_n) = \prod_{i=1}^{n} m_i + \lambda \left(\frac{v-1}{2} - \sum_{i=1}^{n-1} \left(m_i \prod_{j=i+1}^{n} (1 + 2m_i) \right) - m_n \right).$$

With respect to the variable m_1, we have

$$\frac{\partial \psi(m_1, \ldots, m_n)}{\partial m_1} = \prod_{i=2}^{n} m_i - \lambda \prod_{i=2}^{n} (1 + 2m_i) = 0,$$

which implies

$$1 = \lambda \prod_{i=2}^{n} \left(\frac{1}{m_i} + 2 \right). \tag{19}$$

Similarly by partially differentiating of $\psi(m_1, \ldots, m_n)$ with respect to the variable m_2, we have

$$\frac{\partial \psi(m_1, \ldots, m_n)}{\partial m_2} = m_1 \prod_{i=3}^{n} m_i - \lambda \left(2m_1 \prod_{i=3}^{n} (1 + 2m_i) + \prod_{i=3}^{n} (1 + 2m_i) \right) = 0. \tag{20}$$

Substituting (19) into (20) and rearranging it, we have $(2m_2 + 1)m_1 = m_2(2m_1 + 1)$ which implies $m_1 = m_2$. Similarly by partially differentiating of $\psi(m_1, \ldots, m_n)$ with respect to the variable m_3, we have

$$\begin{aligned} \frac{\partial \psi(m_1, \ldots, m_n)}{\partial m_3} =& m_1 m_2 \prod_{i=4}^{n} m_i - \\ & \lambda \left(2m_1(1 + 2m_2) \prod_{i=4}^{n} (1 + 2m_i) + 2m_2 \prod_{i=4}^{n} (1 + 2m_i) + \prod_{i=4}^{n} (1 + 2m_i) \right) = 0. \end{aligned} \tag{21}$$

Substituting (19) and $m_1 = m_2$ into (21) and rearranging it, we have $(2m_3 + 1)m_1 = m_3(2m_1 + 1)$ which implies $m_3 = m_1$.

Now let us use induction to prove that $m_1 = m_2 = \cdots = m_n$. Assume that we have $m_1 = m_2 = \cdots = m_k$ where $3 \leq k < n - 1$. Now we will show $m_{k+1} = m_1$ by partially differentiating of $\psi(m_1, \ldots, m_n)$ with respect to the variable m_{k+1}, we have

$$\begin{aligned} \frac{\partial \psi(m_1, \ldots, m_n)}{\partial m_{k+1}} =& \prod_{i \in [n] \setminus \{k+1\}} m_i - \lambda \left(2 \sum_{j=1}^{k-1} \left(m_j \prod_{h \in [j+1:n] \setminus \{k+1\}} (1 + 2m_h) \right) + \right. \\ & \left. 2m_k \prod_{i=k+2}^{n} (1 + 2m_i) + \prod_{i=k+2}^{n} (1 + 2m_i) \right) = 0. \end{aligned} \tag{22}$$

Submitting the results $m_1 = m_2 = \cdots = m_k$ into (22), we have

$$\begin{aligned} m_1^k \prod_{j=k+2}^{n} m_j =& \lambda \prod_{i=k+2}^{n} (1 + 2m_i) \left(2m_1 \sum_{j=0}^{k-1} (1 + 2m_1)^j + 1 \right) \\ =& (1 + 2m_1)^k \lambda \prod_{i=k+2}^{n} (1 + 2m_i). \end{aligned} \tag{23}$$

In addition when $m_1 = m_2 = \cdots = m_k$, (19) can be written as follows.

$$\frac{1}{\left(\frac{1}{m_1}+2\right)^{k-1}\left(\frac{1}{m_{k+1}}+2\right)} = \lambda \prod_{i=k+2}^{n}\left(\frac{1}{m_i}+2\right). \tag{24}$$

Submitting the results (24) into (23), we have

$$\begin{aligned} & m_1^k = (1+2m_1)^k \frac{1}{\left(\frac{1}{m_1}+2\right)^{k-1}\left(\frac{1}{m_{k+1}}+2\right)} \\ \iff \quad & m_{k+1}\,(1+2m_1) = m_{k+1}(1+2m_1) \iff m_{k+1} = m_1. \end{aligned}$$

When $k = n$ we can also show that our statement holds by the above method.

By the above discussion, we have the optimal solution $m_1 = m_2 = \cdots = m_n$ of the optimization problem (12). From (11) and (12) we have

$$\sum_{i=1}^{n-1}\left(m_i \prod_{j=i+1}^{n}(1+2m_i)\right) + m_n = m_1 \sum_{i=0}^{n-1}(1+2m_1)^i = \frac{(1+2m_1)^n - 1}{2} \le \frac{v-1}{2},$$

which implies $m_1 = m_2 = \cdots = m_n \le \frac{\sqrt[n]{v}-1}{2}$. So the total number of blocks of NHSDP is $b \le (\frac{\sqrt[n]{v}-1}{2})^n$ and the memory ratio is at least $1 - \frac{(\sqrt[n]{v}-1)^n}{v}$. Moreover, when $\sqrt[n]{v} := q$ for some odd integer $q \ge 3$, (9) can be written as follows.

$$\mathcal{D}_{\mathbf{a}} = \left\{ \sum_{i=1}^{n} \alpha_i a_i q^{i-1} \;\middle|\; \alpha_i \in \{-1,1\}, i \in [n] \right\}, \quad \forall \mathbf{a} = (a_1, a_2, \ldots, a_n) \in [q]^n. \tag{25}$$

REFERENCES

[1] C. Abdel Nour and C. Douillard, "Improving bicm performance of qam constellations for broadcasting applications," in *Proc. Int. Symp. on Turbo Codes and Related Topics*, Sep. 2008, pp. 55–60.

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

[3] 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.

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

[5] 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.

[6] K. Wan, D. Tuninetti, and P. Piantanida, "An index coding approach to caching with uncoded cache placement," *IEEE Trans. Inf. Theory*, vol. 66, no. 3, pp. 1318–1332, Mar. 2020.

[7] 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.

[8] 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.

[9] 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.

[10] 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.

[11] 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.

[12] 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.

[13] J. Wang, M. Cheng, K. Wan, and G. Caire, "Placement delivery array construction via cartesian product for coded caching," *IEEE Transactions on Information Theory*, vol. 69, no. 12, pp. 7602–7626, 2023.

[14] J. Wang, M. Cheng, Y. Wu, and X. Li, "Multi-access coded caching with optimal rate and linear subpacketization under pda and consecutive cyclic placement," *IEEE Transactions on Communications*, vol. 71, no. 6, pp. 3178–3190, 2023.

[15] X. Wu, M. Cheng, L. Chen, C. Li, and Z. Shi, "Design of coded caching schemes with linear subpacketizations based on injective arc coloring of regular digraphs," *IEEE Transactions on Communications*, vol. 71, no. 5, pp. 2549–2562, 2023.

[16] X. Zhong, M. Cheng, and R. Wei, "Coded caching schemes with linear subpacketizations," *IEEE Transactions on Communications*, vol. 69, no. 6, pp. 3628–3637, 2021.

[17] J. Li and Y. Chang, "Placement delivery arrays based on combinatorial designs," *IEEE Communications Letters*, vol. 26, no. 2, pp. 296–300, 2022.

[18] E. Peter, K. K. Krishnan Namboodiri, and B. Sundar Rajan, "Shared cache coded caching schemes with known user-to-cache association profile using placement delivery arrays," in *2022 IEEE Information Theory Workshop (ITW)*, 2022, pp. 678–683.

[19] X. Wu, M. Cheng, C. Li, and L. Chen, "Design of placement delivery arrays for coded caching with small subpacketizations and flexible memory sizes," *IEEE Transactions on Communications*, vol. 70, no. 11, pp. 7089–7104, 2022.

[20] V. R. Aravind, P. K. Sarvepalli, and A. Thangaraj, "Lifting constructions of pdas for coded caching with linear subpacketization," *IEEE Transactions on Communications*, vol. 70, no. 12, pp. 7817–7829, 2022.

[21] 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.

[22] 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.

[23] 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.

[24] 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.

[25] D. Katyal, P. N. Muralidhar, and B. S. Rajan, "Multi-access coded caching schemes from cross resolvable designs," *IEEE Trans. Commun.*, vol. 69, no. 5, pp. 2997–3010, May 2021.

[26] 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.

[27] S. Agrawal, K. V. Sushena Sree, and P. Krishnan, "Coded caching based on combinatorial designs," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Jul. 2019, pp. 1227–1231.

[28] M. Xu, Z. Xu, G. Ge, and M.-Q. Liu, "A rainbow framework for coded caching and its applications," *IEEE Transactions on Information Theory*, vol. 70, no. 3, pp. 1738–1752, 2024.

[29] A. A. Mahesh and B. Sundar Rajan, "A coded caching scheme with linear sub-packetization and its application to multi-access coded caching," in *Proc. IEEE Inf. Theory Workshop (ITW)*, Apr. 2021, pp. 1–5.

[30] S. Sasi and B. S. Rajan, "Multi-access coded caching scheme with linear sub-packetization using pdas," *IEEE Transactions on Communications*, vol. 69, no. 12, pp. 7974–7985, 2021.

[31] J. Yin, "Some combinatorial constructions for optical orthogonal codes," *Discret. Math.*, vol. 185, no. 1-3, pp. 201–219, 1998. [Online]. Available: https://doi.org/10.1016/S0012-365X(97)00172-6

[32] T. C. Brown and J. P. Buhler, "A density version of a geometric ramsey theorem," *Journal of Combinatorial Theory, Series A*, vol. 32, no. 1, pp. 20–34, 1982.

[33] N. Alon and M. Naor, "Derandomization, witnesses for boolean matrix multiplication and construction of perfect hash functions," *Algorithmica*, vol. 16, no. 4, pp. 434–449, 1996.

[34] S. Jukna, *Extremal combinatorics: with applications in computer science*. Springer, 2011, vol. 571.

[35] C. Shangguan and G. Ge, "Separating hash families: A johnson-type bound and new constructions," *SIAM Journal on Discrete Mathematics*, vol. 30, no. 4, pp. 2243–2264, 2016. [Online]. Available: https://doi.org/10.1137/15M103827X

[36] C. Elsholtz, Z. Hunter, L. Proske, and L. Sauermann, "Improving behrend's construction: Sets without arithmetic progressions in integers and over finite fields," *arXiv preprint arXiv:2406.12290*, 2024.

[37] R. Fuji-Hara, "Perfect hash families of strength three with three rows from varieties on finite projective geometries," *Designs, Codes and Cryptography*, vol. 77, no. 2, pp. 351–356, 2015.

[38] 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.

[39] 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.

[40] D. Stinson, "Cryptography theory and practice, chapman &hall/crc," 2006.

[41] C. J. Colbourn, *CRC handbook of combinatorial designs*. CRC press, 2010.

[42] S. R. Blackburn, "Combinatorics and threshold cryptography," in *Combinatorial Designs and their Applications*. Routledge, 2023, pp. 49–70.

[43] S. R. Blackburn, M. Burmester, Y. Desmedt, and P. R. Wild, "Efficient multiplicative sharing schemes," in *International Conference on the Theory and Applications of Cryptographic Techniques*. Springer, 1996, pp. 107–118.

[44] C. J. Colbourn, S. S. Martirosyan, T. Van Trung, and R. A. Walker, "Roux-type constructions for covering arrays of strengths three and four," *Designs, Codes and Cryptography*, vol. 41, pp. 33–57, 2006.

[45] A. Fiat and M. Naor, "Broadcast encryption," in *Advances in Cryptology — CRYPTO' 93*, D. R. Stinson, Ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 1994, pp. 480–491.

[46] M. Cheng, J. Jiang, H. Li, Y. Miao, and X. Tang, "Bounds and constructions for 3 3¯-separable codes with length 3," *Designs, Codes and Cryptography*, vol. 81, pp. 317–335, 2016.

[47] C. Shangguan and G. Ge, "Separating hash families: A johnson-type bound and new constructions," *SIAM Journal on Discrete Mathematics*, vol. 30, no. 4, pp. 2243–2264, 2016.