This article has been accepted for publication in IEEE Transactions on Information Theory. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TIT.2024.3352020

A rainbow framework for coded caching and its applications

Min Xu, Zixiang Xu, Gennian Ge and Min-Qian Liu

***Abstract*—The centralized coded caching focuses on reducing the network burden in peak times in a wireless network system. In this paper, motivated by the study of the only rainbow 3-term arithmetic progressions set, we propose a combinatorial framework for constructing coded caching schemes. This framework builds bridges between coded caching schemes and lots of combinatorial objects due to the freedom of the choices of families and binary operations. We prove that any scheme based on a placement delivery array (PDA) can be represented by a rainbow scheme under this framework and lots of other known schemes can also be included in this framework. Moreover, we also present a new coded caching scheme with linear subpacketization and near constant rate using the only rainbow 3-term arithmetic progressions set. Finally, we modify the framework to be applicable to the coded caching problem in Device-to-Device (D2D) networks and the distributed computing problem.**

***Index Terms*—Coded caching scheme, D2D caching, combinatorial framework, only rainbow arithmetic progressions set.**

I. Introduction

In recent years, with the increasing popularity of content-centric wireless communication systems such as 5G, wireless data traffic has become a major challenge in our daily lives. The exponential growth in demand for video content is a significant driving factor for this increase. Moreover, the high temporal variability of network traffic results in communication systems that are congested during peak-traffic times but underutilized during off-peak times. One solution to reduce peak traffic is to take advantage of memories distributed across the network to duplicate content. This process, called *caching*, is performed during off-peak times when network resources are abundant. During peak-traffic times, user demands can be served from these caches, thereby reducing network congestion. *Coded caching*, which makes use of the coding method, can further reduce the transmission.

Coded caching schemes typically involve two phases: *the placement phase* at off peak times and *the delivery phase* at peak times after receiving the demands from active users. In a *centralized coded caching system*, there is a central server which has access to a library of N independent files with the same size and K users each of which has a cache with size of M files, where $M < N$. The server and users are connected by an error-free shared link, that is, all messages sent by the server can be received by all users without error. We refer the bits of messages transmitted during the delivery phase as *communication load* and the *rate* of a caching scheme is the communication load normalized by the size of files.

The study of designing caching schemes that take advantage of coding was initiated by Maddah-Ali and Niesen [24]. In their groundbreaking work, they proposed a centralized coded caching scheme, that involves designing an appropriate content placement and delivery strategy. Although their proposed scheme was proved to be near-optimal, there is a practical disadvantage, that is, the scheme requires each file to be divided into F packets, where F grows exponentially with the number of users K. This is referred to as the *subpacketization level*. A higher subpacketization level implies that the size of files stored by the central server will be large for even moderate number of users, since each file must be split into F pieces and higher indexing overheads are needed to identify the subfiles. Subsequently, a critical aspect of designing centralized coded caching schemes has been focused on reducing the subpacketizations while keeping the rate of the scheme low.

Recently, several interesting methods were used to construct coded caching schemes with low subpacketization levels [1], [3]–[10], [21], [26], [30], [32], [33], [35], [36], [43], [44], [46], [51]. In particular, Yan, Cheng, Tang and Chen [44] represented the *placement delivery array* (PDA) framework and the coded caching scheme they proposed has significantly lower subpacketizations than that of the Maddah-Ali and Niesen scheme in [24]. In [30], Shangguan, Zhang and Ge established a connection between coded caching schemes and 3-partite 3-uniform $(6,3)$-free hypergraphs and constructed coded caching schemes with constant rate and subpacketizations increasing sub-exponentially with the number of users. This connection was further expanded upon in terms of strong edge colorings of bipartite graphs by Yan, Tang, Chen and Cheng [46]. In [33], Shanmugam, Tulino and Dimakis showed coded caching with linear subpacketizations and near constant rate is possible using Ruzsa-Szemerédi graphs. There are also some other combinatorial approaches such as linear block codes [35], [36], line graphs of bipartite graphs [6]–[8], [10], [21] and combinatorial designs [1].

In [24], the authors showed that the delivery phase of the

This project was supported by the National Key Research and Development Program of China under Grant 2020YFA0712100 and Grant 2018YFA0704703, the National Natural Science Foundation of China under Grant 11971325, Grant 12131001, Grant 12371260 and Grant 12231014, Beijing Scholars Program, the Institute for Basic Science (IBS-R029-C4), and National Ten Thousand Talents Program.

M. Xu (minxu0716@qq.com) and M. Liu (mqliu@nankai.edu.cn) are with NITFID, LPMC & KLMDASR, School of Statistics and Data Science, Nankai University,Tianjin 300071, China.

Z. Xu (zxxu8023@qq.com) is with Extremal Combinatorics and Probability Group, Institute for Basic Science, Daejeon, South Korea. This work was started when Z. Xu is a Ph.D student in Capital Normal University.

G. Ge (gnge@zju.edu.cn) is with the School of Mathematical Sciences, Capital Normal University, Beijing 100048, China.

coded caching problem can be viewed as an index coding problem with fixed content placement and demands. Index coding methods are commonly used to evaluate the performance of caching and construct schemes with optimal rate for fixed caching ratio M/N [18], [38], [40]. Achieving the optimal rate typically requires high subpacketization levels, as seen in [24]. In this paper, we propose a combinatorial scheme with a fixed subpacketization level, and leverage the idea of local coloring in the index coding problem to further reduce the rate of scheme. Additionally, our scheme is applicable to D2D networks, where coded caching has been explored in [14]–[16], [29], [48]. In *D2D coded caching problem* [14], the cache placement phase is identical to that of traditional caching. During the delivery phase, after a user requests a file, the server becomes inactive, and the users exchange subfiles they possess to enable each user to decode its requested file based on its own cache and the information transmitted by other users. A scheme which is order optimal within a constant factor when the memory size is large is proposed in [14]. Similar with [24], the scheme also suffers from the high level of subpacketization. In [50], Zhang, Yang and Ji consider a design framework with optimal rate and less subpacketization. In [39], Wang, Cheng, Yan and Tang modified the PDA scheme to D2D networks.

The coded caching problem has been extended to several practical settings over the years, enhancing its applicability. For instance, decentralized schemes proposed in [25] eliminate the need for a central server in the system. Additionally, schemes have been proposed in [12], [49] to handle non-uniform file popularity. The coded caching problem has also been studied in other network scenarios, such as online coded caching [28], [45], hierarchical coded caching [19], and multiple access [12], [17] or servers [34]. These extensions are motivated by practical scenarios where traditional coded caching schemes may not be sufficient or suitable.

The objective of this paper is to present a novel coded caching scheme with a linear subpacketization level. As we have mentioned above, utilizing various combinatorial structures to construct caching schemes is an effective approach. Many of these known schemes seamlessly fit within the PDA framework. Our objective is to explore the specific properties requisite for a combinatorial structure to serve as a foundation for constructing an effective caching scheme. In this paper, we employ coloring as our key combinatorial tool. Assuming the presence of a coded caching scheme with *symmetric uncoded placement*, that is, each user caches subfiles directly instead of caching functions of subfiles, and if a user caches j-th segment of one file, it caches the j-th segment of all files. Subsequently, we analyze this scheme from the perspective of coloring as outlined below. First, we treat the users and the subpacketizations as two sets. Then, the subfiles, which are sent in the delivery phase, can be labeled by the elements in the product of these two sets. For instance, if user i requires file W_{d_i} and does not cache the subfile $W_{d_i}^{(j)}$, which is the j-th segment of W_{d_i}, then $W_{d_i}^{(j)}$ will appear in the delivery phase, and it is labeled by the pair (i, j). Note that in the delivery phase of a coded caching scheme, the combinations of subfiles are sent instead of the original subfiles. We map the elements in the product set to the same color if the subfiles labeled by them are sent in the same combination. After the coloring procedure, the uncolored elements are related to the placement phase. In detail, if (i, j) is an uncolored element in the product set, we will know that user i caches the j-th segment of all files. From this perspective, we observe that if the set of colored elements has a special structure, which is referred as the *only rainbow σ-type set* in Section III, we can modify the coloring function such that the decodability of the caching scheme can be ensured. Our proposed scheme is based on a combinatorial concept called *only rainbow arithmetic progressions sets*, which will be defined in Section II. To describe our construction, we introduce a new framework that we call the *rainbow framework*. This framework allows us to describe many existing coded caching schemes in a common manner. Furthermore, we find that this framework is also applicable to the constructions of D2D coded caching schemes as well as coded distributed computing schemes.

Our contribution can be concluded as follows.

1) First, for $K \leq N$, we propose a combinatorial framework to yield centralized coded caching schemes with uncoded placement. In this framework, we first construct two collections which represent the set of users and the set of subpacketizations respectively. Then, we choose a binary operation on these two sets and define a coloring function on the resulting set after this binary operation. After coloring selected elements in the resulting set with specific rules, we show that the uncolored elements can encode storage actions while elements receiving the same color encode delivery actions (XORs), which leads to a simple but very useful relationship between the coded caching scheme and these elements. Furthermore, we employ this novel idea to conduct a review of several existing works and explore potential ways to enhance the performance of these schemes.
2) Next, we study the problem of constructing centralized coded caching schemes with low subpacketization level based on only rainbow 3-term arithmetic progressions sets, which are proposed by Pach and Tomon [27]. We present a coded caching scheme with linear subpacketization and near constant rate. Moreover, we propose a new delivery scheme based on some results in index coding problem, which can further reduce the transmission load.
3) At last, we apply the rainbow framework to D2D coded caching problem, and proposed a scheme which can be viewed as a D2D placement and delivery array. Moreover, this scheme is also applicable to the coded distributed computing (CDC) problem, as there is a close connection between D2D caching and CDC schemes.

The rest of this paper is organized as follows. In Section II, we introduce the models of traditional coded caching, coded caching in the D2D networks and the coded distributed computing, as well as the coloring problem and index coding problem which are the main tools of our new schemes. Motivated by the study of only rainbow arithmetic progressions sets, we propose a generalized rainbow framework in Section III. Surprisingly, we find out that any PDA scheme can be represented in the rainbow framework, and present several examples in Section IV. In Section V, we derive a combinatorial scheme with new parameters based on the only rainbow 3-term arithmetic progressions set. Next, we modify the rainbow framework to be applicable to the D2D coded caching problem and the distributed computing problem in Section VI. At last, we conclude our main results and propose some open problems in Section VII.

II. Preliminary

In this paper, for simplicity, for positive integers n, a and b with $a < b$, we use $[n]$ to denote the set $\{1, 2, \ldots, n\}$ and we use $[a : b]$ to denote the set $\{a, a+1, \ldots, b\}$. We introduce the models of problems which will be investigated in this paper and main tools to construct our schemes in this section.

A. Coded caching

The first problem we study is the coded caching problem, which was first investigated by Maddah-Ali and Niesen in 2014 [24]. In this kind of problem, there is a central server with a library of N files $\{W_1, W_2, \ldots, W_N\}$, each file can be partitioned into F subfiles with equal size. Suppose in this system there are K users, each of which has a cache with size of M files and requires one file in the library. Suppose user i requires file W_{d_i}, denote the *demand vector* as $\mathbf{d} = (d_1, \ldots, d_K)$. Before the users sending their demands $\mathbf{d}$ to the server, the server fills up all the caches. When the server knows the users' demands, it sends $X_{\mathbf{d}}$ according to the users' caches and their demands $\mathbf{d}$. The server and the users are connected by an error-free shared link, that is, every message sent by the server can be seen by all users. The *transmission rate* or just *rate* of this system is defined as

$$R = \max_{\mathbf{d} \in N^K} \frac{|X_{\mathbf{d}}|}{F}.$$

Denote the minimum rate for fixed cache size M as R^*. The main purpose of coded caching is to design the placement of subfiles such that the transmission rate R for any possible user demand vector is as small as possible. In this paper we focus on the caching schemes with *uncoded placement*, that is, each user caches subfiles directly instead of caching functions of subfiles. We use R_u and R_u^* to denote the transmission rate and the minimum rate of coded caching scheme in this case. We always assume that $N \geq K$ in this paper.

In [24], a cut-set bound for R^* was given as follows,

$$R^* \geq \max_{s \in \{1, \ldots, \min\{N, K\}\}} \left(s - \frac{s}{\lfloor N/s \rfloor} M \right).$$

In addition, they proposed a scheme with a rate given by

$$R = K\left(1 - \frac{M}{N}\right) \cdot \frac{1}{1 + \frac{KM}{N}},$$

which has been shown to be optimal under the uncoded placement strategy [37]. Although the rate is optimal, the subpaketization is $F = \exp(K)$, which increases the complexity of the scheme and implies larger file size since each

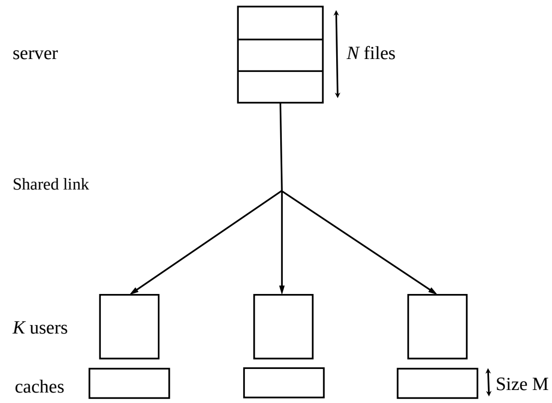

Fig. 1. Coded caching system

file should be split into F pieces, which is exponential in K. Various works about reducing the subpaketizations F while keeping rate R low have been done [1], [3]–[10], [21], [26], [30], [32], [33], [35], [36], [43], [44], [46], [51]. The most general scheme among these works is the placement delivery array (PDA) [44], which can be defined as follows.

Definition II.1. *Given positive integers K, F, M, N with FM/N being an integer, a PDA is an array of size $F \times K$, denoted as $\mathbf{P} = (p_{j,k})_{F \times K}$, which is composed of a specific symbol $*$ and a set of S integers $\mathcal{S} = \{1, 2, \ldots, S\}$. Moreover, the following constrains are required:*

1) *The symbol $*$ appears FM/N times in each column;*
2) *Each integer occurs at least once in the array;*
3) *For any two distinct entries p_{j_1,k_1} and p_{j_2,k_2}, $p_{j_1,k_1} = p_{j_2,k_2} = s$ is an integer only if $j_1 \neq j_2$, $k_1 \neq k_2$ and $p_{j_1,k_2} = p_{j_2,k_1} = *$.*

Example II.1. *The following matrix represents a PDA with $K = 4$, $F = 4$, $\frac{M}{N} = \frac{1}{2}$ and $|S| = 4$.*

$$\begin{pmatrix} * & 3 & 1 & * \\ * & * & 2 & 3 \\ 1 & * & * & 4 \\ 2 & 4 & * & * \end{pmatrix}$$

It is convenient to use this array to represent the placement and the delivery phase of a coded caching scheme, that is, the columns represent users and the rows represent subpacketizations. If $p_{j,k} = *$, then user k caches the j-th segment of each file during the placement phase. If $p_{j,k} = s$, then in the s-th round of delivery, the j-th segment of file W_{d_k} is sent with other subfiles labeled by s. Moreover, the connections between the coded caching schemes and other combinatorial objects were studied in recent years. In [30], the connection of $(6, 3)$-free hypergraphs and PDA schemes implies that if the rate is a constant, the subpacketization level cannot be linear with K. Constructing a scheme with better performance is still an interesting problem.

B. Device-to-Device coded caching and Coded distributed computing

The Device-to-Device (D2D) coded caching problem was originally proposed by Ji, Caire and Molisch in [14]. The model of D2D caching problem is similar to coded caching. This problem shares many similarities with the coded caching problem. In the D2D caching problem, there is a server that is connected to K users through an error-free link. The users are also connected to each other through noiseless device-to-device communication links. The server has access to a library of N files $\{W_1, \cdots, W_N\}$, each file can be split into F subfiles. Each user has a cache of size M. D2D caching system is similar with the server based caching system during the placement phase, that is, the server fills the users' caches without knowing the users' demands. During the delivery phase, each user $k \in [K]$ makes a request for a file W_{d_k}, denote the demand vector as $\mathbf{d} = \{d_1, \cdots, d_k\}$. The server is inactive and each user k sends $\mathbf{X}_{k,\mathbf{d}}$ to other users. After receiving all messages from other users, all users can decode the required files. The transmission rate for D2D caching problem is defined as

$$R_{d2d} := \max_{\mathbf{d}} \sum_{k\in[K]} \frac{|\mathbf{X}_{k,\mathbf{d}}|}{F},$$

and we denote the minimum transmission rate as R^*_{d2d}.

In [14], a converse bound obtained by the method of index coding was given as follows,

$$R^*_{d2d} \geq \max_{l\in\{1,2,\cdots,\min\{N,K\}\}} \left(l - \frac{l}{\lfloor \frac{N}{l} \rfloor}\right).$$

The authors [14] also introduced a scheme that achieves order optimality within a constant factor for large memory sizes. However, similar to [24], this scheme also suffers from high subpacketization levels. To address this issue, Zhang, Yang, and Ji proposed a design framework with optimal rates and less subpacketization in [50]. Additionally, Wang, Cheng, Yan, and Tang modified the PDA scheme for D2D networks in [39]. This D2D placement delivery array can be defined as follows.

Definition II.2. *For positive integers K, F, Z, S, an $F \times K$ array $\mathbf{P} = (p_{j,k}), j \in [F], k \in [K]$, composed of $[S] \cup \{*\}$, is called a (K, F, Z, S) D2D placement delivery array (DPDA) if the following conditions hold.*

1) *The symbol $*$ appears Z times in each column;*
2) *Each integer in $[S]$ occurs at least once in the array;*
3) *For any two distinct entries p_{j_1,k_1} and p_{j_2,k_2}, $p_{j_1,k_1} = p_{j_2,k_2} = s$ is an integer only if $j_1 \neq j_2$, $k_1 \neq k_2$ and $p_{j_1,k_2} = p_{j_2,k_1} = *$.*
4) *There exists a mapping φ from $[S]$ to $[K]$ such that if $p_{j,k} = s$, then $p_{j,\varphi(s)} = *$*

Definition II.2, specifically Condition (4), implies that for any $s \in [S]$, there is a user that caches all subfiles labeled by s. In a recent work of Li and Chang [22], they propose new constructions of DPDA. As an example, we provide an explanation for how D2D caching works.

Example II.2. *Suppose that there are $N = 3$ files denoted as $\{W_1, W_2, W_3\}$, and $K = 3$ users each of which has a cache with size $M = 2$, we can split each file into $F = 6$ pieces with equal size, denoted as $\{W_n^i : i \in [6]\}$. During the placement phase, user 1 caches $\{W_n^i : n \in [3], i \in \{1, 2, 3, 4\}\}$, user 2 caches $\{W_n^i : n \in [3], i \in \{3, 4, 5, 6\}\}$ and user 3 caches $\{W_n^i : n \in [3], i \in \{1, 2, 5, 6\}\}$. During the delivery phase, without loss of generality, we assume user k requires file $W_k, k \in [3]$, that is, the demand vector $\mathbf{d} = (1, 2, 3)$. Then the user 1 sends $\mathbf{X}_{\mathbf{1},\mathbf{d}} = W_3^3 \bigoplus W_2^1$, the user 2 sends $\mathbf{X}_{\mathbf{2},\mathbf{d}} = W_1^6 \bigoplus W_3^4$ and the user 3 sends $\mathbf{X}_{\mathbf{3},\mathbf{d}} = W_1^5 \bigoplus W_2^2$. The decodablity of each user can be checked easily. The rate of this scheme is $\frac{1}{2}$.*

Furthermore, there is another problem in distributed systems known as *coded distributed computing* (CDC), which is closely related to D2D caching [13]. One of the most popular distributed computing frameworks is called *MapReduce*, which was introduced by Dean and Ghemawat in 2008 [11]. In the MapReduce framework, there is a master node responsible for computing Q many functions $\phi_1, \ldots, \phi_Q$ on N files $W_1, \ldots, W_N$ with the help of K worker nodes that can store files and perform computations. We assume that all files have the same size, and each function can be decomposed into map functions and reduce functions as

$$\phi_j(W_1, \ldots, W_N) = f_j(g_{j1}(W_1), \ldots, g_{jN}(W_N)),$$

where $\{g_{jn} : j \in [Q], n \in [N]\}$ are map functions and $\{f_j : j \in [Q]\}$ are reduce functions. We set $v_{jn} := g_{jn}(W_n) \in \mathbb{F}_2^T$ to be the intermediate value of the computing task.

The MapReduce model contains three phases: map, shuffle and reduce. In the map phase, the master node assigns N files to K distributed worker nodes. Each worker node stores a subset of files $\mathcal{M}_k \subset \{W_i, i \in [N]\}$. The computation load r is the average number of worker nodes storing each file, i.e.

$$r = \frac{\sum_{k\in[K]} |\mathcal{M}_k|}{N}.$$

Worker node k computes $\{v_{jn} : j \in [Q], n \in \mathcal{M}_k\}$, which means the node computes intermediate values based on the files it has. In the shuffle phase, all worker nodes need to communicate with each other to get the files they do not have. Each worker node will be assigned Q/K functions $\mathcal{M}_k = \{\phi_{k_1}, \ldots, \phi_{k_{Q/K}}\}$. To finish the computation tasks, each node will send a function of intermediate values it has to other nodes based on the function assignment. We use X_k to denote the message sent by node k and $|X_k|$ is the size of X_k. Then, the communication load L is defined as

$$L = \frac{\sum_{k\in[K]} |X_k|}{QNT}.$$

In the reduce phase, after receiving all messages from other nodes, worker node k can get all intermediate values to finish the computation. We use an example of word counting to illustrate the Mapreduce model as follows.

Example II.3. *Consider the problem of counting $Q = 3$ specific words such as $\{$"and''", "the''", "of''"$\}$ in a book with $N = 6$ chapters denoted as $\{W_n : n \in [6]\}$. In this case, $\phi_j, j = 1, 2, 3$ correspond to the counts of words "and", "the","of" respectively and $g_{jn}, j \in \{1, 2, 3\}, n \in [6]$*

This article has been accepted for publication in IEEE Transactions on Information Theory. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TIT.2024.3352020

corresponds to the counts of words in each chapter. Suppose there are $K = 3$ worker nodes, each of which can store 4 chapters of the book. For instance, worker node 1 stores $\mathcal{M}_1 = \{W_n : n \in \{1,2,3,4\}\}$, worker node 2 stores $\mathcal{M}_2 = \{W_n : n \in \{3,4,5,6\}\}$ and worker node 3 stores $\mathcal{M}_3 = \{W_n : n \in \{1,2,5,6\}\}$. Therefore, the computation load $r = \frac{3\times 4}{6} = 2$. In the map phase, each node computes $g_{jn}, j = 1,2,3$ on its assigned subfiles $\mathcal{M}_i$. In the reduce phase, each node is assigned to find the overall count in the book of one specific word, e.g., suppose worker node 1 need to count the word "and" in the book. It is not difficult to find that

$$\phi_1(W_1, W_2, \ldots, W_6) = f_1(g_{11}(W_1), g_{12}(W_2), \ldots, g_{16}(W_6)),$$

where f_1 is the sum function of the counts of "and" on each chapter. After the map phase, worker node 1 already knows the number of "and" and two other words in W_1, W_2, W_3, W_4. In order to obtain enough information for the reduce phase, worker node 1 needs the value $g_{15}(W_5), g_{16}(W_6)$. In the shuffle phase, the three nodes exchange their information as follows. The node 1 sends $g_{21}(W_1)+g_{33}(W_3)$, the node 2 sends $g_{16}(W_6)+g_{34}(W_4)$ and the node 3 sends $g_{15}(W_5)+g_{22}(W_2)$. Then each node can decode the information it needs to finish the computation. The communication load in this scheme is $L = \frac{3}{6} = \frac{1}{2}$.

It is obvious that the CDC scheme in Example II.3 is equivalent to the D2D caching scheme in Example II.2. In general, in CDC problem, if $Q = K$, then the CDC scheme with N files, K worker nodes and computation load r is equivalent to the D2D caching scheme with K files each of which is split into $F = N$ pieces, K users and cache size $M = r$. Li, Maddah-Ali, Yu and Avestimehr [23] studied the tradeoff between L and r and proposed a scheme attaining the optimal tradeoff. The scheme requires that the number of files N grows exponentially with K. Therefore, reducing the number of files is also an important goal in CDC problem. Yan, Wigger, Yang and Tang [47] used PDA to construct a distributed computing scheme which has a larger communication load and a smaller number of files. Almost at the same time, Konstantinos and Ramamoorthy [20] used another combinatorial object to construct a scheme with $L = \frac{1}{r-1}(1 - \frac{r}{K})$ and $N = \mathrm{Poly}(K)$.

C. Index coding

Index coding is a topic that is closely related to coded caching and information theory. The problem deals with the challenge of broadcasting messages to users who have some prior knowledge, or side information. In the index coding problem, a sender aims to communicate N independent messages to K users over a noiseless channel. Each user $k \in [K]$ demands a set of messages indexed by $D_j \subseteq [N]$ and owns a set of messages $S_j \subseteq [N]$ as side information. It is assumed that $S_j \cap D_j = \emptyset$, and that $D_j \neq \emptyset$ and $S_j \neq [N]$. The goal of the index coding problem is to determine the minimum number of bits of messages that the sender needs to send to satisfy the demands of all the users.

In particular, when the set D_j contains only one element, the index coding problem can be represented using a directed graph known as the *interference graph* G_d. The interference graph G_d consists of n vertices, where each vertex x_i corresponds to user i who requires message x_i. We refer x_j as side information of user i since user i will use these messages it has to decode the desired message. In coded caching problems, we can similarly call the subfiles cached by a user as the side information it has. For this specific index coding problem, there exists a scheme with a transmission rate $R = \chi_l(\bar{G}_d)$, as described in [31], where $\chi_l(\bar{G}_d)$ is the local chromatic number of the complementary graph $\bar{G}_d$. In the directed graph $\bar{G}_d$, the closed out-neighborhood of a given vertex i is denoted as:

$$N^+(i) = \{j \in V(\bar{G}_d) : (i,j) \in E(\bar{G}_d)\} \cup \{i\}.$$

Here, $V(\bar{G}_d)$ denotes the set of vertices in the complementary graph $\bar{G}_d$, and $E(\bar{G}_d)$ denotes the set of edges in $\bar{G}_d$.

Definition II.3 ([31])**.** *The local chromatic number of a directed graph $\bar{G}_d$ is the maximum number of colors in any out-neighborhood minimized over all proper colorings of the undirected graph obtained by ignoring the orientation of the edges in $\bar{G}_d$, i.e.*

$$\chi_l(\bar{G}_d) = \min_c \max_{i\in V} |c(N^+(i))|,$$

where c runs over all proper colorings of the undirected graph.

Next, we explain the index coding problem and the scheme using local chromatic number as follows.

Example II.4. *Consider the index coding problem described by the directed graph G_d, and the complementary graph $\bar{G}_d$ in Figure 2. There are 5 users and 5 messages. The user i demands x_i and owns $x_{i+2}, x_{i+3}, x_{i+4}$ for each $i \in [5]$. From the definition of local chromatic number, we have $\chi_l(\bar{G}_d) = 2$. Then the sender can send*

$$\begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \end{pmatrix} \begin{pmatrix} x_1 + x_4 \\ x_2 + x_5 \\ x_3 \end{pmatrix} = \begin{pmatrix} x_1 + x_4 + x_3 \\ x_2 + x_5 + x_3 \end{pmatrix}.$$

It is easy to check that each user can decode the message it demands.

D. Only rainbow arithmetic progressions set

A *k-term arithmetic progression* in $[n]$ is a sequence $a_1, a_2, \ldots, a_k$ such that $a_i - a_{i-1}, \forall i \in \{2,\ldots,k\}$ is a constant. For instance, $1,3,5,7$ is a 4-term arithmetic progression. We start with the definition of only rainbow arithmetic progressions set. For convenience, we usually write k-term arithmetic progressions as k-APs.

Definition II.4 (Only rainbow k-APs set)**.** *Let A be a subset of $[n]$, and ϕ be a coloring function which colors every element of A. Say A is an only rainbow k-APs set if all k-APs in A are rainbow, that is, for any given k-AP in A, the elements of this k-AP receive distinct colors.*

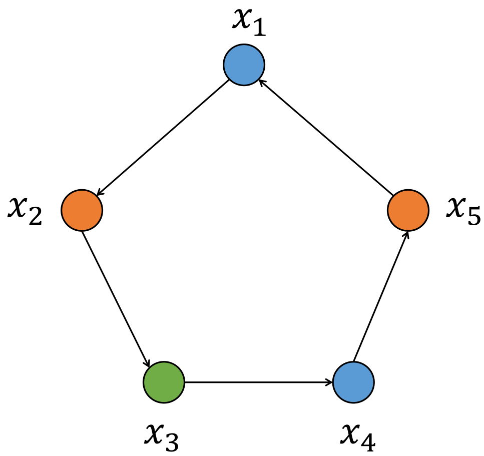

Fig. 2. Example of index coding with local coloring.

Example II.5. *Let $n = 8$, $A = \{2, 3, 4, 6, 7, 8\} \subseteq [8]$, and ϕ be a coloring function over A defined as follows:*

$$\phi(x) = \begin{cases} a, & x = 2, 8, \\ b, & x = 3, 7, \\ c, & x = 4, 6. \end{cases}$$

Then, it is easy to check that the elements of any 3-AP in A receive distinct colors.

How many colors do we need to make sure that every k-AP is rainbow, that is, all of its elements receive distinct colors? For example, if $k = 3$, then at least $\frac{n}{2}$ colors are needed. Instead of coloring the whole set $[n]$, very recently, Pach and Tomon [27] considered the problem on dense subsets of $[n]$. They gave a surprising result that much fewer colors suffice if we do not insist on coloring all elements in $[n]$. More precisely, they showed the following result for $k = 3$.

Theorem II.1 ([27])**.** *Let C be a sufficiently large integer and $n = C^d$ for some positive integer d. There is a set $A \subseteq [n]$ with $|A| \geqslant n - n^\alpha$ and a coloring of A with n^β colors such that every 3-AP in A is rainbow, where $\alpha = 1 - \frac{1}{18C^6 \log C}$ and $\beta = \log_C(10C^{\frac{16}{C}} \log_2 C)$.*

For convenience, we call set A an (α, β)-only rainbow 3-APs set if A satisfies the properties in Theorem II.1. Moreover, Theorem II.1 can be extended to longer arithmetic progressions easily. For more details, we refer the readers to Concluding remarks in [27].

III. Generalized rainbow framework

We will describe a combinatorial framework for coded caching schemes with uncoded placement when $K \leq N$. Let $\mathcal{A}$ and $\mathcal{B}$ be two collections with $K = |\mathcal{A}|$ and $F = |\mathcal{B}|$. Let $\uplus$ be a certain binary operation, for example, the binary operation $\uplus$ can be a simple addition operation or a set union operation. Then we define the set $\mathcal{C}$ as

$$\mathcal{C} = \mathcal{A} \biguplus \mathcal{B} = \{a \biguplus b : a \in \mathcal{A}, b \in \mathcal{B}\}.$$

Suppose $\hat{\mathcal{C}}$ is a subset of $\mathcal{C}$. For a carefully selected property σ, we call any subset in $\mathcal{C}$ with property σ as a *σ-type structure*. Consider a coloring function ϕ defined on $\hat{\mathcal{C}}$, a subset S of $\hat{\mathcal{C}}$ is called *rainbow* if no two distinct elements in S receive the same color. We call $\hat{\mathcal{C}}$ *only rainbow σ-type set* if every σ-type structure in $\hat{\mathcal{C}}$ is rainbow under the coloring function ϕ. Moreover, we need another coloring function $\Phi : \{(a, b) : a \in \mathcal{A}, b \in \mathcal{B}, a \uplus b \in \hat{\mathcal{C}}\} \mapsto \{c_1, c_2, \ldots, c_l\}$, which is based on ϕ. For example, when $\mathcal{A} = \mathcal{B} = [m]$ and the binary operation is addition, the σ-type structure can be 3-term arithmetic progressions.

In our framework, when we design a coloring function Φ to color every pair (a, b) with a single color, the following condition is required.

Condition III.1. *For any sets $\mathcal{A}, \mathcal{B}$, the binary operation $\uplus$ and the selected subset $\hat{\mathcal{C}} \subseteq \mathcal{C}$, the σ-type structure, the coloring functions ϕ and Φ should satisfy that if $\Phi(a, b) = \Phi(a', b')$, then $a \uplus b', a' \uplus b \notin \hat{\mathcal{C}}$.*

Definition III.1. *Let $\mathcal{A}, \mathcal{B}$ be two sets with size K and F respectively. Let $\uplus$ be a binary operation, $\mathcal{C} = \mathcal{A} \uplus \mathcal{B}$ and $\hat{\mathcal{C}} \subseteq \mathcal{C}$. Suppose there exists a triple (σ, ϕ, Φ) such that under the coloring function ϕ, every σ-type structure in $\hat{\mathcal{C}}$ is rainbow, and Condition III.1 is satisfied. Then we can obtain a coded caching scheme with N files and K users, where the users are labeled by the elements in $\mathcal{A}$ and the files are denoted as $\{W_n : n \in [N]\}$. Suppose each file has a size of B bits, and user a has a cache with a capacity of $M_a \cdot B$ bits, where*

$$\frac{M_a}{N} = \frac{|\{b : a \uplus b \notin \hat{\mathcal{C}}\}|}{F}.$$

The placement and delivery phase can be described as follows:

1) ***Placement phase:*** *For each file W_n, split it to F subfiles with equal size and label them with elements in $\mathcal{B}$, i.e., $W_n = \{W_n^{(b)} : b \in \mathcal{B}\}$. For uncolored elements $a \uplus b$ in $\mathcal{A} \uplus \mathcal{B}$, user a caches the b-th packet of all files in the library, that is, $\{W_n^{(b)} : n \in [N], b \in \mathcal{B}, a \uplus b \notin \hat{\mathcal{C}}\}$.*
2) ***Delivery phase:*** *For any demand vector $\mathbf{d} = (d_{a_1}, \ldots, d_{a_k})$, the delivery is based on the coloring function Φ. Suppose there are s elements $(a_1, b_1), (a_2, b_2), \ldots, (a_s, b_s)$ receiving the same color c from Φ, then we denote the following XOR multiplexing of packets as W_c*

$$W_c = \bigoplus_{1 \leqslant i \leqslant s} W_{d_{a_i}}^{(b_i)}.$$

For each uncached pair (a, b), define a constant

$$m(a, b) = \#\{\Phi(a', b') : \ a' = a \text{ or } b' \biguplus a \in \hat{\mathcal{C}}\},$$

and $m = \max\{\{m(a, b) : a \uplus b \in \hat{\mathcal{C}}\} \cup \{|\Phi| - 1\}\}$. Let P be an $m \times |\Phi|$ maximum distance separable (MDS) matrix. During the delivery phase, the server sends

$$\mathbf{X}_d = P \cdot (W_{c_1}, W_{c_2}, \ldots, W_{c_{|\Phi|}})^T.$$

Thus, the delivery phase consists of m packet transmissions.

The main motivation of above framework is as follows. The sets $\mathcal{A}$ and $\mathcal{B}$ represent the users and the subpacketizations, respectively. We use $\hat{\mathcal{C}}$ to describe the placement phase, that is, if $a \biguplus b \notin \hat{\mathcal{C}}$, the user a caches subfiles $\{W_n^{(b)} : n \in [N]\}$. The coloring function Φ tells us which symbols can be sent together. If each symbol appears at most once in the delivery phase, from the decodability of the caching scheme, we can derive Condition III.1. Finally, we further reduce the rate combining some ideas from the index coding.

Theorem III.1. *The scheme in Definition III.1 is decodable.*

Proof: Denote the demand vector as $\mathbf{d} = (d_{a_1}, \ldots, d_{a_K})$, for any user a_i, we need to show that user a_i can decode the subfile $W_{d_{a_i}}^{(b)}$ for all $b \in \mathcal{B}$ such that $a_i \biguplus b \in \hat{\mathcal{C}}$ from $\mathbf{X_d}$. Without loss of generality, we can assume that $\Phi(a_i, b) = c_1$, by Condition III.1, we know that the user a_i can decode $W_{d_{a_i}}^{(b)}$ from W_{c_1} since user a_i caches $\{W_n^{(b')} : n \in [N], a_i \biguplus b' \notin \hat{\mathcal{C}}\}$. For the pair (a_i, b), define a set

$$S(a_i, b) := \{\Phi(a', b') : \ a' = a_i \text{ or } b' \biguplus a_i \in \hat{\mathcal{C}}\}.$$

Suppose that the size of $S(a_i, b)$ is $m' \leq m$, where m is defined in Definition III.1, let $S(a_i, b) = \{c_1, c_2, \ldots, c_{m'}\}$. The central server sends $\mathbf{X_d} = P \cdot (W_{c_1}, \ldots, W_{c_l})^T$, where $P = (p_1, p_2, \ldots, p_l)$ is an $m \times l$ MDS array and p_i is the i-th column of P. After receiving $\mathbf{X_d}$, the user a_i can delete all subfiles it has and obtain

$$(p_1, p_2, \ldots, p_{m'})(W_{d_{a_i}}^{(b)}, W'_{c_2}, \ldots, W'_{c_{m'}})^T,$$

where W'_{c_j} is obtained by deleting some redundant components in W_{c_j}. Since for any color $c \notin S(a_i, b)$, the user a_i caches all the subfiles in W_c and for any color $c \in S(a_i, b) \backslash \{c_1\}$, there exists at least one pair (a', b') such that the user a_i does not cache. Because P is an MDS array, any m columns of P are linearly independent and $m' \leq m$, then the user a_i can decode $W_{d_{a_i}}^{(b)}$. ∎

Remark III.1. *In the delivery phase, we define a constant m and find an $m \times |\Phi|$ MDS array. To make sure that such an array exists over $\mathbb{F}_2$ (which is always considered in coded caching problem), m cannot be smaller than $|\Phi| - 1$. However, if we assume that the computation between subfiles is over $\mathbb{F}_q$ for sufficiently large q, then we don't need $m \geq |\Phi| - 1$, which means we can further reduce the transmission rate. From Definition II.3, it is obvious that the constant m in the delivery scheme is the local chromatic number of the corresponding index coding problem.*

Fix $a \in \mathcal{A}$, denote Z_a as the number of $b \in \mathcal{B}$ such that $a \biguplus b \in \hat{\mathcal{C}}$. In particular, if each user has the same cache size, i.e. $Z_a = Z$ for all $a \in \mathcal{A}$, we have the following result.

Corollary III.1. *The generalized rainbow framework provides a coded caching scheme with $(K = |\mathcal{A}|, F = |\mathcal{B}|, \frac{M}{N} = 1 - \frac{Z}{F}, R = \frac{m}{|\mathcal{B}|})$.*

To better illustrate the scheme, we present an example as follows.

Example III.1. *Let $\mathcal{A} = [4]$, $\mathcal{B} = \{12, 23, 34, 41\}$ and the binary operation be set union operation $\cup$. Therefore, $\mathcal{C} = \mathcal{A} \cup \mathcal{B}$ and assume $\hat{\mathcal{C}} = \{123, 124, 134, 234\}$. The coloring function ϕ on $\hat{\mathcal{C}}$ satisfies the rainbow property such that any three elements in $\hat{\mathcal{C}}$ receive distinct colors, which implies there are exactly 4 colors. Using ϕ, we can define the coloring function Φ over all the pairs (a, b) such that $\Phi(a, b) = \phi(a \cup b)$.*

During the delivery phase, we need to count a special number before sending messages. For any pair $(a, b) \in \mathcal{A} \times \mathcal{B}$ such that $a \cup b \in \hat{\mathcal{C}}$, define $m(a, b) = \#\{\Phi(a', b') : \ a' = a \text{ or } b' \cup a \in \hat{\mathcal{C}}\}$ and $m = \max_{\{(a,b): a \cup b \in \hat{\mathcal{C}}\}} m(a, b)$. In this example, $m = 3$. Next, find a 3×4 MDS array such as

$$P = \begin{pmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \end{pmatrix},$$

and define the sum of subfiles corresponding to the same color c_i as

$$W_{c_i} = \bigoplus_{\Phi(A_i \cup B_i) = c_i} W_{d_{a_i}}^{(b_i)}.$$

Then we send

$$P \cdot (W_{c_1}, W_{c_2}, W_{c_3}, W_{c_4})^T.$$

More precisely, suppose $\phi(123) = c_1$, $\phi(124) = c_2$, $\phi(134) = c_3$, $\phi(234) = c_4$, then we send

$$W_{d_1}^{(23)} \bigoplus W_{d_3}^{(12)} \bigoplus W_{d_2}^{(34)} \bigoplus W_{d_4}^{(23)},$$
$$W_{d_2}^{(14)} \bigoplus W_{d_4}^{(12)} \bigoplus W_{d_2}^{(34)} \bigoplus W_{d_4}^{(23)},$$
$$W_{d_1}^{(34)} \bigoplus W_{d_3}^{(14)} \bigoplus W_{d_2}^{(34)} \bigoplus W_{d_4}^{(23)}.$$

We can check that each user can decode all the subfiles he requires from the transmission.

The scheme defined in Definition III.1 is a valid coded caching scheme even for heterogeneous cache size. Note that Condition III.1 implies that if the central server sends messages according to colors, i.e., $\mathbf{X_d} = (W_{c_1}, \ldots, W_{c_l})$, it is a PDA scheme with heterogeneous cache size. The MDS array P can be used to further compress the transmission since for each user, there are at most m colors c such that the user does not cache all the components of W_c.

Remark III.2. *In fact, if we take advantage of the coloring function Φ to color every pair (a, b) with a single color, and the central server sends by colors, then Condition III.1 is necessary. Since each uncached pair (a, b) is sent only once, the user a needs to decode it from the transmission implies that any other subfile $W_{d_{a'}}^{(b')}$ combined with $W_{d_a}^{(b)}$ must be already cached by the user a. In this case, the generalized scheme defined above is equivalent to combining a PDA scheme with a local coloring. However, if the coloring function Φ is a multi-coloring function, that is, it can color some pair (a, b) with multiple colors, this scheme may perform better than the PDA scheme. In this case, Condition III.1 is not necessary. We provide an example to show that if Φ is a multi-coloring function, then our framework is able to cover some caching schemes which can not be covered by the simple combination of PDA schemes and local coloring method.*

In the following example, we write $\{a_1, a_2, \ldots, a_s\}$ as $a_1 a_2 \cdots a_s$ for ease and we use $[i : j]_6$ to denote the set $\{i, i+1, i+2, \ldots, j\} \mod 6$.

This article has been accepted for publication in IEEE Transactions on Information Theory. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TIT.2024.3352020

Example III.2. *Let* $\mathcal{A} = \{12, 23, 34, 45, 56, 61\}, \mathcal{B} = \{1234, 2345, 3456, 4561, 5612, 6123\}$. *Let* $\biguplus$ *be the set union operation. Then we can see that* $\mathcal{C}$ *is the union of cyclic sets with size* $4, 5, 6$. *Next we choose* $\hat{\mathcal{C}}$ *as follows:*

$$\begin{aligned}\hat{\mathcal{C}} &= \{12345, 23456, 34561, 45612, 56123, 61234, 123456\}\\ &= \{[i : i+4]_6 : i \in [6]\} \cup \{123456\}.\end{aligned}$$

Let ϕ *be a coloring function such that for any* $[i : i+4]_6$, *if* $C \in \hat{\mathcal{C}}$ *contains it, then* C *should receive a unique color, i.e.,*

$$\phi(C) = \begin{cases} c_i, & C = [i : i+4]_6,\\ \{c_1, c_2, c_3, c_4, c_5, c_6\}, & C = 123456.\end{cases}$$

And Φ *is defined as*

$$\Phi(a,b) = \begin{cases} \phi(a \cup b), & \text{if } |a \cup b| = 5,\\ \{c_1, c_3, c_5\}, & \text{if } |a \cup b| = 6 \text{ and}\\ & \quad a = \{k, k+1\},\ k \text{ odd},\\ \{c_2, c_4, c_6\}, & \text{if } |a \cup b| = 6 \text{ and}\\ & \quad a = \{k, k+1\},\ k \text{ even}.\end{cases}$$

The coded caching scheme can be described as follows: The set $\mathcal{A}$ *consists of* $K = 6$ *users and each file in* $\{W_n, n \in [N], N \geq 6\}$ *can be partitioned into* $F = 6$ *subfiles, that is,* $W_n = \{W_n^{[i:i+3]_6} : i \in [6]\}$. *During the placement phase, the user* a *caches* W_n^b *if* $a \subseteq b$, *that is,*

User 12 caches $\{W_n^{5612}, W_n^{6123}, W_n^{1234} : n \in [N]\}$;
User 23 caches $\{W_n^{6123}, W_n^{1234}, W_n^{2345} : n \in [N]\}$;
User 34 caches $\{W_n^{1234}, W_n^{2345}, W_n^{3456} : n \in [N]\}$;
User 45 caches $\{W_n^{2345}, W_n^{3456}, W_n^{4561} : n \in [N]\}$;
User 56 caches $\{W_n^{3456}, W_n^{4561}, W_n^{5612} : n \in [N]\}$;
User 61 caches $\{W_n^{4561}, W_n^{5612}, W_n^{6123} : n \in [N]\}$.

During the delivery phase, suppose the demand of user $\{i, i+1\}$ *is file* i, *the central server can send by colors, that is,*

$$\begin{aligned}&W_1^{2345} \bigoplus W_4^{1234} \bigoplus W_1^{3456} \bigoplus W_3^{5612} \bigoplus W_5^{1234};\\ &W_2^{3456} \bigoplus W_5^{2345} \bigoplus W_2^{4561} \bigoplus W_4^{6123} \bigoplus W_6^{2345};\\ &W_3^{4561} \bigoplus W_6^{3456} \bigoplus W_1^{3456} \bigoplus W_3^{5612} \bigoplus W_5^{1234};\\ &W_4^{5612} \bigoplus W_1^{4561} \bigoplus W_2^{4561} \bigoplus W_4^{6123} \bigoplus W_6^{2345};\\ &W_5^{6123} \bigoplus W_2^{5612} \bigoplus W_1^{3456} \bigoplus W_3^{5612} \bigoplus W_5^{1234};\\ &W_6^{1234} \bigoplus W_3^{6123} \bigoplus W_2^{4561} \bigoplus W_4^{6123} \bigoplus W_6^{2345}.\end{aligned}$$

The decodability of above scheme can be easily checked. For example, user 12 needs W_1^{2345}, W_1^{3456} *and* W_1^{4561}. *After eliminating the subfiles cached by user 12, we will obtain*

$$\begin{pmatrix} 1&1&0&0&0&0&0&0&0\\ 0&0&0&1&1&0&1&1&0\\ 0&1&0&0&0&1&0&0&1\\ 0&0&1&0&1&0&0&1&0\\ 0&1&0&0&0&0&0&0&0\\ 0&0&0&0&1&0&0&1&0 \end{pmatrix} \cdot \begin{pmatrix} W_1^{2345}\\ W_1^{3456}\\ W_1^{4561}\\ W_2^{3456}\\ W_2^{4561}\\ W_3^{4561}\\ W_5^{2345}\\ W_6^{2345}\\ W_6^{3456} \end{pmatrix}.$$

Since the first three columns are linearly independent with the other six columns, the user 12 *can decode the three subfiles he wants. The decodability of other users can be shown similarly.*

Note that if we use PDA scheme for the caching problem in Example III.2, there are at least 8 transmissions. Suppose we use local coloring as well, from Definition III.1, there are at least 7 transmissions since the MDS array over $\mathbb{F}_2$ with 8 columns contains at least 7 rows. Therefore, the above scheme cannot be represented by PDA or the combination of PDA and local coloring.

The main idea of Example III.2 comes from another method in the index coding problem named as *interference alignment*. We call a set of subfiles in the transmission as *interferences* for one user if the user does not require these subfiles and does not cache them. To be specific, in above example, the interferences for user 12 are $W_2^{3456}, W_2^{4561}, W_3^{4561}, W_5^{2345}, W_6^{2345}, W_6^{3456}$. The interference alignment method is to make the coefficient space of interferences independent with the coefficient space of the demand files for each user. This method can be transformed into the constrains for the coloring function Φ. To see this, we need some notations.

For a pair $(a, b), a \biguplus b \in \hat{\mathcal{C}}$ and a coloring function Φ with l colors, denote the *color set* assigned to (a, b) by Φ as $C(a, b)$, that is, $C(a, b) = \Phi(a, b) \subseteq \{c_1, \ldots, c_l\}$. An *indicator vector* $\mathbf{v}(a, b) = (v_1, v_2, \ldots, v_l)$ corresponding to the color set $C(a, b)$ is a binary vector with length l, where $v_i = 1$ if $c_i \in C(a, b)$ and $v_i = 0$ otherwise.

From above analysis, if Φ can map some pairs (a, b) with multiple colors, we can assume the following condition holds.

Condition III.2. *For any sets* $\mathcal{A}, \mathcal{B}$, *the binary operation* $\biguplus$ *and the selected subset* $\hat{\mathcal{C}} \subseteq \mathcal{C}$, *the* σ*-type structure, the coloring functions* ϕ *and* Φ *should satisfy that for any* $a \in \mathcal{A}$, $\{\mathbf{v}(a, b) : b \in \mathcal{B}, a \biguplus b \in \hat{\mathcal{C}}\}$ *is a set of linearly independent vectors and* $\{\mathbf{v}(a, b) : b \in \mathcal{B}, a \biguplus b \in \hat{\mathcal{C}}\}$ *is linearly independent with the vector set* $\{\mathbf{v}(a', b) : a \biguplus b \in \hat{\mathcal{C}}, a' \biguplus b \in \hat{\mathcal{C}}\}$.

Remark III.3. *Given user set* $\mathcal{A}$ *and packets set* $\mathcal{B}$, *the first task in our generalized rainbow framework is to define the suitable binary operation* $\biguplus$, *and then the most important thing is to give the appropriate* σ*-type structure and coloring functions* ϕ, Φ. *From above analysis, the above conditions are restricted to* Φ. *The* σ*-type structure and coloring function such that every* σ*-type structure is rainbow can be chosen widely. This might bring some new ideas in constructing coded caching schemes. Trivially, we can color every element of* $\hat{\mathcal{C}}$ *with different colors, which implies that arbitrary structure in* $\hat{\mathcal{C}}$ *is rainbow.*

IV. Existing schemes under rainbow framework

In this section, we will highlight several existing works on coded caching schemes via different ideas and combinatorial objects. However, as we have discussed in Remark III.3, in our generalized rainbow framework, many existing schemes are equipped with the trivial coloring function. Hence we represent some of them and discuss the way to improve the existing schemes.

A. All PDA schemes are rainbow schemes

For a bipartite graph, a strong edge coloring function is a coloring function such that any two edges which have the same color are not adjacent and can not be connected by another edge. The relationship between PDA schemes and strong edge coloring of a bipartite graph is studied in [46]. The following result is known.

Theorem IV.1 ([46]). *Any $F\times K$ array P is a PDA if and only if its corresponding edge colored bipartite graph $G(\mathcal{F}\cup\mathcal{K},\mathcal{E})$ satisfies*

1) *the vertex in $\mathcal{K}$ has a constant degree;*
2) *the corresponding coloring is a strong edge coloring.*

Given a bipartite graph G with vertex set $\mathcal{F}\cup\mathcal{K}$ and edge set $\mathcal{E}$. Then, in our new framework, let $\mathcal{A}=\mathcal{K}$, $\mathcal{B}=\mathcal{F}$ and the binary operation be the Cartesian product, i.e.

$$\mathcal{C}=\mathcal{A}\times\mathcal{B}=\{(a,b):a\in\mathcal{A},b\in\mathcal{B}\}.$$

Note that $\mathcal{C}$ is the set of all possible edges in G. Choose $\hat{\mathcal{C}}=\mathcal{E}$, and define a coloring function ϕ on $\hat{\mathcal{C}}$ such that

1) If $(a_1,b_1),(a_2,b_2)\in\hat{\mathcal{C}}$ and $a_1=a_2$ or $b_1=b_2$ then $\phi((a_1,b_1))\neq\phi((a_2,b_2))$.
2) If $(a_1,b_1),(a_2,b_2),(a_3,b_3)\in\hat{\mathcal{C}}$ and $|\{a_1,a_2,a_3\}|\leq 2$, $|\{b_1,b_2,b_3\}|\leq 2$ then $(a_1,b_1),(a_2,b_2)$ and (a_3,b_3) are rainbow.

In fact, any strong edge coloring function satisfies above conditions. It is easy to check that the first condition means that any two edges with the same color are not adjacent and the second condition is equivalent to that any two edges with the same color cannot be connected by another edge. Then we can select the coloring function Φ in the rainbow framework exactly equal to ϕ. If we do not consider the constant m and send subfiles according to their colors directly, we obtain a PDA scheme. From above analysis, we know that any PDA scheme can be represented by a rainbow scheme under the rainbow framework.

Remark IV.1. *In the delivery phase, if we take m into consideration, then the transmission rate can be further reduced since $m\leq|\Phi|$.*

B. Construction from the union of disjoint subsets

The first construction from [30] regards users and packets as disjoint subsets of the ground set, respectively. More precisely, they set

$$\left(K,F,\frac{M}{N},R\right)=\left(\binom{n}{a},\binom{n}{b},\frac{\binom{n}{b}-\binom{n-a}{b}}{\binom{n}{b}},\frac{\binom{n}{a+b}}{\binom{n}{b}}\right).$$

In particular, for $a=2$, $n=\lambda a$ for constant $\lambda>1$, this construction achieves $R=\lambda^2$ with $F=O(K^{-\frac{1}{4}}\cdot 2^{\sqrt{2K}H(\lambda^{-1})})$, where $H(x)=-x\log_2 x-(1-x)\log_2(1-x)$ for $0<x<1$ is the binary entropy function. Moreover, it is easy to check R and $\frac{M}{N}$ are both constant and F grows sub-exponentially with K under such parameters. This can actually be achieved by subset version of rainbow schemes as follows.

Definition IV.1. *Let $\mathcal{A}=\binom{[n]}{a}$ be a collection of all a-element subsets of $[n]$ and $\mathcal{B}=\binom{[n]}{b}$ be a collection of all b-element subsets of $[n]$. Set the binary operation $\biguplus$ as set union operation $\bigcup$. Suppose that a and b are positive integers with $a<b$, and n is large enough, then it is easy to see that*

$$\mathcal{C}=\mathcal{A}\biguplus\mathcal{B}=\binom{[n]}{b}\bigcup\binom{[n]}{b+1}\bigcup\cdots\bigcup\binom{[n]}{a+b}.$$

Let $\hat{\mathcal{C}}\subseteq\mathcal{C}$ be the collection of all $(a+b)$-element subsets of $[n]$, i.e. $\hat{\mathcal{C}}=\binom{[n]}{a+b}$. It is easy to see $|\hat{\mathcal{C}}|=|\mathcal{C}|-o(|\mathcal{C}|)$. We then color every element in $\hat{\mathcal{C}}$ using the proper coloring function Φ, and leave the elements in $\mathcal{C}\setminus\hat{\mathcal{C}}$ uncolored.

1) ***Placement phase:*** *For uncolored elements $A\bigcup B$ in $\mathcal{A}\biguplus\mathcal{B}$, user A caches the B-th packet of all files in the library.*
2) ***Delivery phase:*** *The delivery is based on the coloring function Φ. Suppose there are s elements $A_1\bigcup B_1,A_2\bigcup B_2,\ldots,\ A_s\bigcup B_s$ receiving the same color from Φ, then the server broadcasts the following XOR multiplexing of packets*

$$\bigoplus_{1\leqslant i\leqslant s}W_{d_{A_i}}^{(B_i)}.$$

The delivery phase consists of $|\Phi|$ packet transmissions, where $|\Phi|$ is the number of colors in Φ.

It remains to discuss the properties of the proper coloring function Φ. Let $A_1,A_2\in\mathcal{A}$ and $B_1,B_2\in\mathcal{B}$. Using the proof of Theorem V.1, neither of the following will happen.

- $A_1\bigcup B_1$ and $A_1\bigcup B_2$ cannot receive the same color from Φ. Also, $A_2\bigcup B_1$ and $A_2\bigcup B_2$ cannot receive the same color from Φ.
- If $A_1\bigcup B_1$ and $A_2\bigcup B_2$ receive the same color from Φ, then both of $A_1\bigcup B_2$ and $A_2\bigcup B_1$ are uncolored.

To satisfy the above conditions, we can design the coloring functions with the following properties.

Lemma IV.1. *Let Φ be the coloring function of elements in $\hat{\mathcal{C}}=\binom{[n]}{a+b}$ with the following properties:*

- *If $C_1,C_2\in\hat{\mathcal{C}}$ and $|C_1\bigcap C_2|\geqslant a$, then $\Phi(C_1)\neq\Phi(C_2)$.*
- *If $C_1,C_2,C_3\in\hat{\mathcal{C}}$ and $C_i\subseteq C_j\bigcup C_k$ for $i\neq j\neq k$, then C_1,C_2 and C_3 receive distinct colors.*

Then Φ can be used in the above set system version of rainbow scheme.

Remark IV.2. *However, there is only a trivial coloring function Φ yet, that is, we color each element in $\hat{\mathcal{C}}=\binom{[n]}{a+b}$ via different colors. This trivial coloring function achieves the scheme of [30]. Hence, any proper coloring function with fewer than $\binom{n}{a+b}$ colors will improve this construction.*

C. Ali-Niesen scheme

As far as we know, Ali-Niesen scheme [24] is the first coded caching scheme that kickstarted the research of coded caching in general. Recall the parameters in Maddah Ali-Niesen scheme as

$$\left(K,F,\frac{M}{N},R\right)=\left(K,\binom{K}{\frac{KM}{N}},\frac{M}{N},\frac{K(1-\frac{M}{N})}{\frac{KM}{N}+1}\right).$$

This article has been accepted for publication in IEEE Transactions on Information Theory. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TIT.2024.3352020

Using the generalized rainbow scheme, we can set $\mathcal{A} = [n]$ and $\mathcal{B} = \binom{[n]}{t}$, where $t = \frac{KM}{N}$. Using the similar analysis, it suffices to design the coloring functions with properties as follows.

Lemma IV.2. *Let $t = \frac{KM}{N}$ and Φ be the coloring function of elements in $\hat{\mathcal{C}} = \binom{[n]}{t+1}$ such that*

- *If $C_1, C_2 \in \hat{\mathcal{C}}$ and $C_1 \bigcap C_2 \neq \emptyset$, then $\Phi(C_1) \neq \Phi(C_2)$.*
- *If $C_1, C_2, C_3 \in \hat{\mathcal{C}}$ and $C_i \subseteq C_j \bigcup C_k$ for $i \neq j \neq k$, then C_1, C_2, C_3 receive distinct colors.*

Then Φ can be used in the Ali-Niesen type rainbow scheme.

D. Tang-Ramamoorthy scheme

In the Tang-Ramamoorthy scheme [35], the subpacketization is exponentially smaller than that of the previous scheme but with some minor loss in rate (up to a constant factor). However, $F = e^{f(\frac{N}{M})\cdot K}$ for some function $f(\cdot)$. We show an example which can also be achieved by our generalized rainbow scheme. Assume that there exists a generator matrix G of an (n, k) linear block code over field $\mathbb{F}_q$ which has the following properties:

1) **Divisibility:** $(k+1) \mid n$,
2) **Rank property:** For every contiguous set of $k+1$ columns, every subset of k-columns on this $k+1$ subset has full rank.

Then we can obtain q^k codewords of length n.

We use $\mathcal{A}$ to denote all the pairs (a, a'), where $a \in [n]$ and $a' \in \{0, 1, \dots, q-1\}$. Each codeword $b_1 b_2 \dots b_n$ corresponds to the set $\{(1, b_1), (2, b_2), \dots, (n, b_n)\}$. Therefore, all of the q^k codewords correspond to q^k sets, which form the family $\mathcal{B}$.

Definition IV.2. *Let $\mathcal{A}$ and $\mathcal{B}$ be the families defined above. Let $\uplus$ be the set union operation $\bigcup$ and $\mathcal{C} = \mathcal{A} \cup \mathcal{B}$. It is easy to show that for any $C \in \mathcal{C}$, C contains exactly n or $n+1$ pairs. Let $\hat{\mathcal{C}} \subseteq \mathcal{C}$ be the collection of elements in $\mathcal{C}$ which contains exactly $n+1$ pairs. It is easy to see $|\hat{\mathcal{C}}| = |\mathcal{C}| - o(|\mathcal{C}|)$. We then color every element using the proper coloring function, and leave the elements in $\mathcal{C} \setminus \hat{\mathcal{C}}$ uncolored. The placement phase and delivery phase are the same as those in Definition III.1.*

Now, we give a coloring function when $n = k+1$. Any $\{C_1, \dots, C_{k+1}\} \subseteq \hat{\mathcal{C}}$ satisfying

$$|C_i \cap C_j| = n, \text{ for any } i \neq j \in [k+1], \tag{1}$$

forms a color class. Next we explain such a coloring function in detail.

For each sequence $s_1 s_2 \dots s_n$ which is not a codeword, its corresponding set $\{(1, s_1), \dots, (n, s_n)\} \notin \mathcal{B}$. Due to the rank property, for each $j \in [n]$, there is a unique codeword $b_1 b_2 \dots b_n$ such that $b_i = s_i$ for all $i \in [n] \setminus \{j\}$. Therefore, for $j \in [n]$, there is a unique element B_j in $\mathcal{B}$ such that $\{(k, s_k) | k \in [n] \setminus j\} \subseteq B_j$. We color $B_j \cup (j, s_j) \in \hat{\mathcal{C}}$ for every $j \in [k+1]$ with the same color, thus the condition (1) is satisfied, where the set of n common pairs corresponds to the sequence $s_1 s_2 \dots s_n$.

Example IV.1. *Suppose $q = 2$, $k = 2$, $n = k+1 = 3$, we have $K = 6$, $F = 4$. The generator matrix G of $(3, 2)$ block code is*

$$\begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix}.$$

Then $\mathcal{A} = \{(1,0), (1,1), (2,0), (2,1), (3,0), (3,1)\}$, $\mathcal{B} = \{B_1, B_2, B_3, B_4\}$, where

$$\begin{aligned} B_1 &= \{(1,0), (2,0), (3,0)\}, \\ B_2 &= \{(1,0), (2,1), (3,1)\}, \\ B_3 &= \{(1,1), (2,0), (3,1)\}, \\ B_4 &= \{(1,1), (2,1), (3,0)\}. \end{aligned}$$

Using the procedure above, we can color the elements in $\hat{C}$ with 4 colors.

$$\begin{aligned} c_1 = \{&\{(1,1), (1,0), (2,0), (3,0)\}, \\ &\{(2,0), (1,1), (2,1), (3,0)\}, \\ &\{(3,0), (1,1), (2,0), (3,1)\}\}; \\ c_2 = \{&\{(2,1), (1,0), (2,0), (3,0)\}, \\ &\{(1,0), (1,1), (2,1), (3,0)\}, \\ &\{(3,0), (1,0), (2,1), (3,1)\}\}; \\ c_3 = \{&\{(3,1), (1,0), (2,0), (3,0)\}, \\ &\{(1,0), (1,1), (2,0), (3,1)\}, \\ &\{(2,0), (1,0), (2,1), (3,1)\}\}; \\ c_4 = \{&\{(3,1), (1,1), (2,1), (3,0)\}, \\ &\{(1,1), (1,0), (2,1), (3,1)\}, \\ &\{(2,1), (1,1), (2,0), (3,1)\}\}. \end{aligned}$$

Finally, it achieves a $(K, F, \frac{M}{N}, R) = (6, 4, \frac{1}{2}, 1)$ centralized coded caching scheme.

V. New rainbow schemes for coded caching

A. New rainbow schemes

In this section, we introduce our new scheme for the centralized coded caching problem under the rainbow framework in the previous section. Suppose there are K users served through a noiseless broadcast channel by an agent who has access to N distinct files from a library. Every user is equipped with a local cache of size M. The key problem is to design the *placement phase* where the user caches file packets from the library under the cache constraint and the *delivery phase* where the user reveals his own demand so that all demands should be satisfied with at most R file transmissions.

Let $\mathcal{W} = \{W_1, W_2, \dots, W_N\}$ be a library of N files. Let $(W_i^{(1)}, W_i^{(2)}, \dots, W_i^{(F)}) \in \mathbb{F}^{F \times 1}$ be a vector of length F over some field $\mathbb{F}$ representing file W_i. Then we recall the (R, K, M, N, F) centralized coded caching scheme as follows.

Definition V.1. *Every file W_i in the library is divided into F packets for $1 \leqslant i \leqslant N$. An (R, K, M, N, F) centralized coded caching scheme consists of:*

1) *A family of subsets $\{W_{i,j}\}_{i \in [N], j \in [K]}$, where $W_{i,j} \subseteq [K]$ is the set of user caches where the i-th packet of file j is stored. Moreover, each user can cache at most MF file packets in placement phase.*

2) *A set of user demands $\boldsymbol{d} = (d_1, d_2, \ldots, d_K)$ arising from the library, where $d_i \in [N]$ is the index of the requested file of the user k. The transmission function $\phi(W_{d_1}, W_{d_2}, \ldots, W_{d_K}) \to \mathbb{F}^{RF}$ for some field $\mathbb{F}$ such that every user s can decode their demanded files W_{d_s} via ϕ and the cache content available.*
3) *For any demand pattern among the users arising from the library, the total number of file transmission can be at most R.*

Definition V.2 (Rainbow coded caching scheme)**.** *Let $K = m$ be an integer. Let $A \subseteq [2m]$ be an (α, β)-only rainbow 3-APs set. More precisely, $A \subseteq [2m]$ is a set of size at least $2m-(2m)^\alpha$, and let χ be a coloring of A with at most $(2m)^\beta$ colors such that every 3-AP in A is rainbow. Let $A_1 = A_2 = [m]$, we consider the following sum set*

$$A_1 + A_2 = \{x + y : x \in A_1, y \in A_2\}.$$

We then color the pairs $(x, y) \in A_1 \times A_2$ as

$$\Psi((x,y)) = \begin{cases} \textit{uncolored}, & x+y \notin A, \\ (x-y, \chi(x+y)), & x+y \in A. \end{cases}$$

Then we describe the placement phase and delivery phase with assistance of the above colored sum set. In this scheme, every file in the library is split into $F = K$ packets.

1) ***Placement phase:*** *For uncolored elements $x+y$ in $A_1 + A_2$, user x caches the y-th packet of all files in the library.*
2) ***Delivery phase:*** *The delivery is based on the coloring function Ψ. Suppose that there are s elements $x_1+y_1, x_2+y_2, \ldots, x_s+y_s$ receiving the same color from Ψ, then the server broadcasts the following XOR multiplexing of packets*

$$\bigoplus_{1 \leqslant i \leqslant s} W_{d_{x_i}}^{(y_i)}.$$

The delivery phase consists of $|\Psi|$ packet transmissions, where $|\Psi|$ is the number of colors in Ψ.

Remark V.1. *Note that in this new scheme, we omit the constant m defined in the framework and send messages according to their colors directly. In this scheme, A_1 and A_2 stand for the sets of users and subfiles. The arithmetic sum of x and y determines the placement phase and the coloring function Φ determines the delivery phase.*

Next we show that our rainbow scheme is a centralized coded caching scheme.

Theorem V.1. *The (K, α, β, Ψ) rainbow scheme is an $(R = \frac{|\Psi|}{F}, K, M, N, F = K)$ coded caching scheme.*

Proof. In our rainbow scheme, the number of packets per file F is equal to the number of users K. Our first task is to verify that the cache constraint of every user is satisfied. Note that for every user $x \in [m]$, there are at most $(2m)^\alpha$ elements $b \in [2m] \setminus A$, such that $x + y = b$. This indicates that each user caches at most $(2m)^\alpha N \leqslant MF$ file packets.

Next we will show that our rainbow scheme satisfies any kind of user demands $\mathbf{d} = (d_1, d_2, \ldots, d_K)$ arising from the library, where $d_i \in [N]$ is the index of the requested file of the user i. After coloring each element of sum set $A_1 + A_2$ using function Ψ, we consider some color class $\mathcal{C}_j$ consisting of c_j elements

$$x_1 + y_1, x_2 + y_2, \ldots, x_{c_j} + y_{c_j},$$

and the corresponding XOR transmission consisting of c_j packets:

$$\bigoplus_{1 \leqslant i \leqslant c_j} W_{d_{x_i}}^{(y_i)}.$$

Then we analyze the decoding algorithm for each user. For a user $x \in [m]$ requesting a certain file W_{d_x}, he has cached the set of packets $\{W_{s,d_x} : s + x \text{ is uncolored}\}$ in the placement phase. Hence, to decode the requested file W_{d_x}, it suffices to obtain the uncached packets. We just need to show the following result.

Claim 1. *Let $\mathcal{C}_j$ be some color class consisting of c_j elements*

$$x_1 + y_1, x_2 + y_2, \ldots, x_{c_j} + y_{c_j}.$$

Then for each $1 \leqslant s, t \leqslant c_j$, $x_s + y_t \in A$ if and only if $s = t$.

Proof of Claim 1. Trivially, $x_s + y_t \in A$ if $s = t$ by the definition. On the other hand, if $s \neq t$, suppose $x_s + y_t \in A$, the element $x_s + y_t$ will receive a color from function Ψ. Without loss of generality, let $\Psi(x_s + y_s) = \psi(x_t + y_t) = \Psi_1$ and $\psi(x_s + y_t) = \Psi_2$. Let $\Delta = x_s - y_s = x_t - y_t$, then we can write $x_s + y_t$ as

$$x_s + y_t = \frac{(2x_s - \Delta) + (2y_t + \Delta)}{2} = \frac{(x_s + y_s) + (x_t + y_t)}{2}.$$

Note that $x_s + y_s$, $x_s + y_t$ and $x_t + y_t$ form a 3-AP in A. However, $x_s + y_s$ and $x_t + y_t$ are in the same color class, which implies $\chi(x_s + y_s) = \chi(x_t + y_t)$ by the definition of Ψ. That is impossible since every 3-AP in A is rainbow, then the claim follows. $\square$

By Claim 1, it holds that user x_s knows all the packets $\{W_{d_{x_i}}^{(y_i)} : 1 \leqslant i \leqslant c_j, i \neq s\}$ in his cache at the placement phase. Then the unknown packets can be easily obtained by substraction operation. Every user will recover his requested file by this decoding algorithm, therefore the rainbow scheme works. Finally, it is easy to see that the number of colors used by Ψ is at most $(2m)^{1+\beta}$. This completes the proof of Theorem V.1. $\square$

Next we give an example to show the practicality of our coded caching scheme.

Example V.1. *Suppose $K = F = 4$, we color the elements in $\{1, 2, \ldots, 8\}$ as follows.*

$$\Theta(x) = \begin{cases} \textit{uncolored}, & x = 1, 5, \\ a, & x = 2, 8, \\ b, & x = 3, 7, \\ c, & x = 4, 6. \end{cases}$$

Using the function Ψ in Definition V.2, we present a table as follows. As we can see in Table I, every uncolored pair (x, y) represents a caching action in the placement phase and every color class corresponds to a transmission in delivery phase.

TABLE I
$K = F = 4$ RAINBOW CODED CACHING SCHEME

K \ F	1	2	3	4
1	$(0, a)$	$(-1, b)$	$(-2, c)$	uncolored
2	$(1, b)$	$(0, c)$	uncolored	$(-2, c)$
3	$(2, c)$	uncolored	$(0, c)$	$(-1, b)$
4	uncolored	$(2, c)$	$(1, b)$	$(0, a)$

Finally, it achieves a $(K, F, \frac{M}{N}, R) = (4, 4, \frac{1}{4}, 6)$ *centralized coded caching scheme.*

Obviously, the limitation showed in [30] indicates the following result.

Theorem V.2. *The only rainbow* 3*-APs set with* $n - n^{\alpha}$ *elements and only* $O(1)$ *colors does not exist for any* $0 < \alpha < 1$.

B. Schemes taking m into consideration

Note that in the above rainbow scheme, we first color the non-cached subfiles such that special structure in $\hat{\mathcal{C}}$ is rainbow and then deliver messages depending on their colors. If different colors can be sent together, then the transmission load can be further reduced. In Example III.1, making use of the results of the index coding problem, we calculate a constant m, and use an $m \times |\Phi|$ MDS matrix to combine different colors.

Remark V.2. *In Example III.1, if we deliver the subfiles according to different colors, we have to send messages at* 4 *times. But if we use the new delivery scheme, we only need to send messages at* 3 *times and for each time, the subfiles corresponding to* 2 *colors are used.*

For general case, let $\mathcal{A} = [n]$ and $\mathcal{B} \subset \binom{[n]}{t}$. Set the operator as set union $\cup$. Then $\mathcal{C} = \mathcal{A} \cup \mathcal{B}$ contains some t-tuples and $(t+1)$-tuples of $[n]$. Define $\hat{\mathcal{C}} = \mathcal{C} \cap \binom{[n]}{t+1}$, and the coloring function ϕ over $\hat{\mathcal{C}}$ must satisfy the following rainbow conditions.

- If $C_1, C_2 \in \hat{\mathcal{C}}$ and $C_1 \bigcap C_2 \neq \emptyset$, then $\Phi(C_1) \neq \Phi(C_2)$.
- If $C_1, C_2, C_3 \in \hat{\mathcal{C}}$ and $C_i \subseteq C_j \bigcup C_k$ for $i \neq j \neq k$, then C_1, C_2 and C_3 receive different colors.

The coloring function Φ over $\{(A, B) :\ A \cup B \in \hat{\mathcal{C}}\}$ is defined as

$$\Phi(A, B) = \phi(A \cup B).$$

Suppose that $(A_1, B_1), (A_2, B_2), \ldots, (A_s, B_s)$ are assigned with the same color c, which implies that $A_i \cup B_j \notin \hat{\mathcal{C}}, \forall i \neq j$. Thus user A_i can decode $W_{d_{A_i}}^{(B_i)}$ from $W_c = \bigoplus_{j \in [s]} W_{d_{A_j}}^{(B_j)}$. For each pair (A, B) with $A \cup B \in \hat{\mathcal{C}}$, define a constant

$$m(A, B) = \#\{\Phi(A' \cup B') :\ A' = A \ or\ B' \cup A \in \hat{\mathcal{C}}\},$$

where $(A', B') = (A, B)$ is allowed. Then define

$$m = \max\{\{m(A, B) : A \cup B \in \hat{\mathcal{C}}\} \cup \{|\Phi| - 1\}\}.$$

Let P be an $m \times |\Phi|$ MDS matrix,

$$P = \begin{pmatrix} p_1 & p_2 & \ldots & p_{\Phi} \end{pmatrix}.$$

During the delivery phase, the server sends

$$P \cdot (W_{c_1}, W_{c_2}, \ldots, W_{c_{|\Phi|}})^T.$$

Based on the above construction, we have the following theorem.

Theorem V.3. *The rainbow framework with new delivery scheme described above is an* $(R = \frac{m}{F}, K = |\mathcal{A}|, F = |\mathcal{B}|, M, N)$ *coded caching scheme.*

Proof: It suffices to prove the solvability for each user A_i. From the definition of $\hat{\mathcal{C}}$, we know that user A_i caches the subfile B_j if $A_i \cup B_j \notin \hat{\mathcal{C}}$, and A_i can decode the subfile B_j, which is not cached by the user, from W_{c_1} if $\Phi(A_i, B_j) = c_1$. From the definition of $m(A_i, B_j)$, it is not difficult to see that every subfile $W_{d_{A_m}}^{(B_n)}$ with $A_m \cup B_n \in \hat{\mathcal{C}}$ which is cached by user A_i is not contained in the set

$$\{(A', B') :\ A' = A_i \ or\ B' \cup A_i \in \hat{\mathcal{C}}\}.$$

Thus, for user A_i, after deleting the subfiles he has cached, we have

$$(p_1, p_2, \ldots, p_{m(A_i, B_j)}) \cdot (W_{c_1}, W_{c_2}, \ldots, W_{c_{m(A_i, B_j)}})^T.$$

Since P is an MDS matrix and $m \geq m(A_i, B_j)$, user A_i can decode W_{c_1} and further decode $W_{d_{A_i}}^{(B_j)}$. The similar approach can be used for any (A_i, B_j), thus the solvability of the new delivery scheme is proved. ■

C. Comparison with previous constructions

In Table II, we compare our scheme with some existing schemes with low subpacketization level.

Most schemes contained in Table II are PDA scheme. It is hard to compare these schemes due to the different parameters. The proposed scheme based on the only rainbow 3-AP progressions set is also a PDA scheme. The scheme shown in Example III.1 and discussed in this section combines the PDA scheme with local coloring. If we restrict $\frac{M}{N} = \frac{n-2}{n}$, then the optimal rate achieved by PDA schemes would be 1. In our framework, the rate can be smaller.

VI. Applications to D2D Coded Caching

In this section, we present the application of our rainbow framework in the Device-to-Device coded caching problem. We start with an illustrative example.

Example VI.1. *Let* $\mathcal{A} = [4], \mathcal{B} = \{12, 23, 34, 41\}$. *Suppose there are* 4 *files* $\{W_n : n \in [4]\}$. *Each file can be partition into* 4 *pieces, that is,* $W_n = \{W_n^B : B \in \mathcal{B}\}$ *and there are* 4 *users* $\{K_i : i \in \mathcal{A}\}$. *During the placement phase, the user* K_i *caches* W_n^B *if* $i \in B$. *For example, the user* K_1 *caches subfiles* W_n^{12}, W_n^{41} *for all* $n \in [4]$. *During the delivery phase, suppose the user* K_i *makes a request of file* W_i *for all* $i \in [4]$, *then the transmissions are as follows.*

1) K_1 *sends* $W_2^{41} \bigoplus W_4^{12}$;
2) K_2 *sends* $W_1^{23} \bigoplus W_3^{12}$;
3) K_3 *sends* $W_2^{34} \bigoplus W_4^{23}$;
4) K_4 *sends* $W_1^{34} \bigoplus W_3^{41}$.

Parameters	The Number of users K	Subpacketization F	Caching ratio $\frac{M}{N}$	Transmission rate R
$0<\delta<1$ and c_1, c_2 are constant [33]	K	K	$2K^{-c_1\delta\exp(-\frac{c_2}{\delta})}$	K^{δ}
$n, a, b \in \mathbb{N}^+$ with $n > a+b$ [30]	$\binom{n}{b}$	$\binom{n}{a}$	$1-\frac{\binom{n-b}{a}}{\binom{n}{a}}$	$\frac{\binom{n}{a+b}}{\binom{n}{a}}$
$a, b, m, \lambda \in \mathbb{N}^+$, $a, b < m$, $\lambda < \min\{a, b\}$ [46]	$\binom{m}{a}$	$\binom{m}{b}$	$1-\frac{\binom{a}{\lambda}\binom{m-a}{b-\lambda}}{\binom{m}{b}}$	$\frac{\binom{m}{a+b-2\lambda}\binom{a+b-2\lambda}{a-\lambda}}{\binom{m}{b}}$
$n, w \in \mathbb{N}^+$, $w < n$ [52]	2^n	$\sum_{i=0}^{w}\binom{n}{i}$	$1-\frac{\binom{n}{w}}{\sum_{i=0}^{w}\binom{n}{i}}$	$\frac{\binom{n}{w}2^{n-w}}{\sum_{i=0}^{w}\binom{n}{i}}$
$r, k, z \in \mathbb{N}^+$ [2]	$2^r k$	$2^r k$	$1-\frac{r+1}{2^r}+\frac{rz}{2^r k}$	$\frac{k^2+k^2r-krz}{2^r k}$
$n_0, w, p_0 \in \mathbb{N}^+$ [42]	$p_0^{n_0}$	$p_0^{n_0}$	$1-\frac{\binom{n_0}{w}(p_0-1)^w}{p_0^{n_0}}$	$\frac{\binom{n_0}{w}(p_0-1)^w}{p_0^{n_0-w}}$
$m \in \mathbb{N}^+, \alpha = 1-o(1), \beta = o(1)$	m	m	$2m-(2m)^{\alpha}$	$\frac{(2m)^{1+\beta}}{m}$
$n \in \mathbb{N}^+$, Example III.1	n	n	$\frac{n-2}{n}$	$\frac{n-1}{n}$

TABLE II
COMPARISON WITH SOME EXISTING SCHEMES

It is easy to check that each user can decode all the subfiles it needs. For instance, K_1 can decode W_1^{23} from the message sent by K_2, since W_3^{12} is already known. In this example, the transmission rate $R_{d2d} = \frac{4}{4} = 1$.

Note that in the rainbow scheme for coded caching shown in Example III.1, the transmissions should be

$$W_1^{23} \bigoplus W_3^{12} \bigoplus W_2^{34} \bigoplus W_4^{23},$$
$$W_2^{14} \bigoplus W_4^{12} \bigoplus W_2^{34} \bigoplus W_4^{23},$$
$$W_1^{34} \bigoplus W_3^{14} \bigoplus W_2^{34} \bigoplus W_4^{23}.$$

We can partition each transmission into 2 parts, each part can be sent by only one user. Therefore, we obtain the delivery phase in Example VI.1.

We modify the rainbow framework in Section III to be applicable to D2D coded caching problem as follows.

Definition VI.1. *Let $K = |\mathcal{A}|$, $F = |\mathcal{B}|$, define $\mathcal{C} = \mathcal{A} \uplus \mathcal{B}$. Let the σ-type structure, $\hat{\mathcal{C}}$, and the coloring functions ϕ and Φ be the same as those given in Definition III.1. Then we describe the D2D caching scheme with assistance of the above colored subset $\hat{\mathcal{C}}$.*

1) ***Placement phase:*** *The user a caches $\{W_n^b : a \uplus b \notin \hat{\mathcal{C}}\}$.*
2) ***Delivery phase:*** *The communication between nodes is based on the coloring function Φ. Denote the transmission in Definition III.1 as*

$$P \cdot (W_{c_1}, W_{c_2}, \ldots, W_{c_{|\Phi|}})^T = P \cdot V,$$

where P is an $m \times |\Phi|$ MDS array, and W_{c_i} is the XOR sum of all the subfiles assigned color c_i. Suppose $P \cdot V$ can be decomposed into a linear combination of $W_1', W_2', \ldots, W_{m'}'$, where each W_i' can be contained in the cache of one user i. Then the final transmission is that node i sends W_i' for $i \in [m']$.

In fact, the framework in Definition VI.1 is equivalent to the caching scheme using the PDA for the D2D network (DPDA). Since the condition in the delivery phase of Definition VI.1 is equivalent to the condition (4) in Definition II.2. The framework is rather abstract, we present a specific scheme as follows.

Definition VI.2. *Let $n \geq 4$ be a positive integer, $K = F = n$. Let $\mathcal{A} = [n]$, $\mathcal{B} = \{[i : i+n-3] \mod n : i \in [n]\}, \mathcal{C} = \mathcal{A} \cup \mathcal{B} = \{a \cup b : a \in \mathcal{A}, b \in \mathcal{B}\}$ and $\hat{\mathcal{C}} = \mathcal{C} \cap \binom{[n]}{n-1}$. The coloring function ϕ over $\hat{\mathcal{C}}$ maps all elements to different colors, that is, $\phi(c), c \in \hat{\mathcal{C}}$ are distinct. Let the coloring function $\Phi : \mathcal{A} \times \mathcal{B} \mapsto \{1, 2, \ldots, |\Phi|\}$ satisfy $\Phi(a, b) = \phi(a \cup b)$. Then the scheme can be described as follows:*

1) ***Placement phase:*** *there are $N \geq K$ files $\{W_1, \ldots, W_N\}$ and each file can be split to n pieces, that is, $W_n = \{W_n^b : b \in \mathcal{B}\}$ for any $n \in [N]$. The user i caches $\{W_n^b : b \in \mathcal{B}, i \in b\}$.*
2) ***Delivery phase:*** *suppose the demand vector $\mathbf{d} = (d_1, d_2, \ldots, d_k)$ the user $i+1$ sends*

$$\bigoplus_{k \cup b = [i:i+n-2]} W_{d_k}^b.$$

Theorem VI.1. *The scheme in Definition VI.2 is a D2D coded caching scheme with $F = K = n, \frac{M}{N} = \frac{n-2}{n}, R_{d2d} = 1$.*

Proof: We need to check that in above scheme, each user can decode the subfiles it does not own, that is, for the user a, we need to show that for any $b \in \mathcal{B}$ such that $a \notin b$, then the user a can decode $W_{d_a}^b$ from the transmissions of other users.

Parameters	The Number of users K	Subpacketization F	Caching ratio $\frac{M}{N}$	Transmission rate R
$t, K \in \mathbf{N}^+, t < K$ [14]	K	$t\binom{K}{t}$	$\frac{t}{K}$	$\frac{K}{t} - 1$
integer $q \geq 2$ [41]	q^2	q^q	$\frac{1}{q}$	q
$m, q, z \in \mathbb{N}^+, z < q$ [39]	$mq + q$	$(m+1)\lfloor\frac{q-1}{q-z}\rfloor^2 q^m - \lfloor\frac{q-1}{q-z}\rfloor q^m$	$\frac{z}{q}$	$\frac{(m+1)(q-z)}{(m+1)\lfloor\frac{q-1}{q-z}\rfloor - 1}$
$m, q, z, b \in \mathbb{N}^+,\ z < q, b \leq m$ [39]	$\binom{m}{b} q^b$	$\binom{m}{b}\lfloor\frac{q-1}{q-z}\rfloor^{2b} q^m - \lfloor\frac{q-1}{q-z}\rfloor^b q^m$	$1 - (\frac{q-z}{q})^b$	$\frac{\binom{m}{b}(q-z)^b}{\binom{m}{b}\lfloor\frac{q-1}{q-z}\rfloor^b - 1}$
$n \in \mathbb{N}^+, n \geq 2$	n	n	$\frac{n-2}{n}$	1

TABLE III
COMPARISON WITH SOME EXISTING SCHEMES

First, the user a has file W_b if $a \in b$. In the delivery phase, worker node a can decode $W^b_{d_a}$ from $\bigoplus_{k\cup m=[i:i+n-2]} W^m_{d_k}$ sent by worker node $i+1$ if $a \cup b = [i : i+n-2]$, since

$$\bigoplus_{k\cup m=[i:i+n-2]} W^m_{d_k} = \bigoplus_{k\cup m=[i:i+n-2],(k,m)\neq(a,b)} W^m_{d_k} \bigoplus W^b_{d_a},$$

and $k \cup m = a \cup b$ implies that $a \in m$, that is, the user a owns $W^m_{d_k}$. Therefore, the a can decode all subfiles it requires. ■

We compare the D2D caching scheme in Definition VI.2 with previous schemes in Table III. In addition, as discussed in [13], the D2D coded caching scheme can also be transformed to CDC scheme directly.

VII. Conclusion

In this paper, we investigate the coded caching problem with uncoded placement. Motivated by the study of only rainbow 3-APs sets, we propose a generalized combinatorial framework, which can be applied in D2D networks as well. We observe that any PDA scheme can be represented by a rainbow scheme under our framework, and several existing works can also be included in the framework. Our rainbow framework builds bridges between combinatorial objects and coded caching problems. For any given $\mathcal{A}$, $\mathcal{B}$ and the binary operation $\uplus$, we can obtain a coded caching scheme by selecting a suitable σ-type structure and the coloring function. The freedom of choosing the structure and coloring function enables us to connect more combinatorial objects with coded caching. Moreover, using the idea of the index coding problem, our framework can further reduce the transmission load and obtain some schemes which cannot be represented by a PDA. In addition, if we select the coloring function such that some pairs (a, b) with $a \cup b \in \hat{\mathcal{C}}$ receive multiple colors, then the framework can also contain some examples that cannot be covered by the combination of PDA and local coloring method in index coding.

Next, based on the study of the only rainbow 3-term arithmetic progressions set, we offer a coded caching scheme with linear subpacketization and near constant rate. For several existing works, we propose the corresponding coloring models and we do hope it will be helpful to solve the following problem by designing proper coloring function.

Question VII.1. *Let $\frac{M}{N}$ and R be both constants, prove or disprove the existence of centralized coded caching schemes such that F grows polynomially with K.*

At last, we investigate the application of this rainbow framework in the D2D coded caching. With an additional condition, the framework is equivalent to the D2D placement and delivery array, which is a critical tool to construct D2D caching schemes with uncoded placement. Due to the connection between D2D coded caching and distributed computing, a CDC scheme with F files corresponds to a D2D caching scheme with F subpacketizations. Therefore, a similar question can be asked as following.

Question VII.2. *Let r be a constant. Suppose N cannot be divided by $\binom{K}{r+1}$ or is small than $\exp(K)$, construct schemes which can attain the optimal or suboptimal communication load L.*

References

[1] S. Agrawal, K. V. S. Sree, and P. Krishnan, "Coded caching based on combinatorial designs," in *IEEE International Symposium on Information Theory, ISIT 2019, Paris, France, July 7-12, 2019*, 2019, pp. 1227–1231.

[2] 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.

[3] V. 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.

[4] M. Cheng, J. Jiang, X. Tang, and Q. Yan, "Some variant of known coded caching schemes with good performance," *IEEE Transactions on Communications*, vol. 68, no. 3, pp. 1370–1377, 2019.

[5] M. Cheng, J. Wang, X. Zhong, and Q. Wang, "A framework of constructing placement delivery arrays for centralized coded caching," *IEEE Transactions on Information Theory*, vol. 67, no. 11, pp. 7121–7131, 2021.

[6] H. H. S. Chittoor and P. Krishnan, "Low subpacketization coded caching via projective geometry for broadcast and d2d networks," in *2019 IEEE Global Communications Conference (GLOBECOM)*, 2019, pp. 1–6.

[7] ——, "Projective geometry based coded caching schemes with subexponential and linear subpacketizations," in *19th International Symposium on Communications and Information Technologies, ISCIT 2019, Ho Chi Minh City, Vietnam, September 25-27, 2019*, 2019, pp. 537–542.

[8] H. H. S. Chittoor, P. Krishnan, K. V. S. Sree, and B. Mamillapalli, "Subexponential and linear subpacketization coded caching via projective geometry," *IEEE Trans. Inform. Theory*, vol. 67, no. 9, pp. 6193–6222, 2021.

[9] H. H. S. Chittoor, P. Krishnan, K. S. Sree, and B. Mamillapalli, "Subexponential and linear subpacketization coded caching via projective geometry," *IEEE Transactions on Information Theory*, vol. 67, no. 9, pp. 6193–6222, 2021.

[10] H. H. S. Chittoor, B. MVN, and P. Krishnan, "Coded caching via projective geometry: A new low subpacketization scheme," in *IEEE International Symposium on Information Theory, ISIT 2019, Paris, France, July 7-12, 2019*, 2019, pp. 682–686.

[11] J. Dean and S. Ghemawat, "Mapreduce: simplified data processing on large clusters," *Communications of the ACM*, vol. 51, no. 1, pp. 107–113, 2008.

[12] J. Hachem, N. Karamchandani, and S. N. Diggavi, "Coded caching for multi-level popularity and access," *IEEE Transactions on Information Theory*, vol. 63, no. 5, pp. 3108–3141, 2017.

[13] A. M. Ibrahim, A. A. Zewail, and A. Yener, "Device-to-device coded-caching with distinct cache sizes," *IEEE Transactions on Communications*, vol. 68, no. 5, pp. 2748–2762, 2020.

[14] M. Ji, G. Caire, and A. F. Molisch, "Fundamental limits of caching in wireless d2d networks," *IEEE Transactions on Information Theory*, vol. 62, no. 2, pp. 849–869, 2016.

[15] ——, "Wireless device-to-device caching networks: Basic principles and system performance," *IEEE Journal on Selected Areas in Communications*, vol. 34, no. 1, pp. 176–189, 2016.

[16] M. Ji, R.-R. Chen, G. Caire, and A. F. Molisch, "Fundamental limits of distributed caching in multihop d2d wireless networks," in *2017 IEEE International Symposium on Information Theory (ISIT)*, 2017, pp. 2950–2954.

[17] M. Ji, A. M. Tulino, J. Llorca, and G. Caire, "Caching in combination networks," in *2015 49th Asilomar Conference on Signals, Systems and Computers*. IEEE, 2015, pp. 1269–1273.

[18] ——, "Order-optimal rate of caching and coded multicasting with random demands," *IEEE Transactions on Information Theory*, vol. 63, no. 6, pp. 3923–3949, 2017.

[19] N. Karamchandani, U. Niesen, M. A. Maddah-Ali, and S. N. Diggavi, "Hierarchical coded caching," *IEEE Transactions on Information Theory*, vol. 62, no. 6, pp. 3212–3229, 2016.

[20] K. Konstantinidis and A. Ramamoorthy, "Resolvable designs for speeding up distributed computing," *IEEE/ACM Transactions on Networking*, vol. 28, no. 4, pp. 1657–1670, 2020.

[21] P. Krishnan, "Coded caching via line graphs of bipartite graphs," in *IEEE Information Theory Workshop, ITW 2018, Guangzhou, China, November 25-29, 2018*, 2018, pp. 1–5.

[22] J. Li and Y. Chang, "New constructions of d2d placement delivery arrays," *IEEE Communications Letters*, vol. 27, no. 1, pp. 85–89, 2023.

[23] S. Li, M. A. Maddah-Ali, Q. Yu, and A. S. Avestimehr, "A fundamental tradeoff between computation and communication in distributed computing," *IEEE Transactions on Information Theory*, vol. 64, no. 1, pp. 109–128, 2018.

[24] M. A. Maddah-Ali and U. Niesen, "Fundamental limits of caching," *IEEE Transactions on Information Theory*, vol. 60, no. 5, pp. 2856–2867, 2014.

[25] ——, "Decentralized coded caching attains order-optimal memory-rate tradeoff," *IEEE/ACM Transactions on Networking*, vol. 23, no. 4, pp. 1029–1040, 2015.

[26] J. Michel and Q. Wang, "Placement delivery arrays from combinations of strong edge colorings," *IEEE Transactions on Communications*, vol. 68, no. 10, pp. 5953–5964, 2020.

[27] J. Pach and I. Tomon, "Colorings with only rainbow arithmetic progressions," *Acta Mathematica Hungarica*, vol. 161, no. 2, pp. 507–515, 2020.

[28] R. Pedarsani, M. A. Maddah-Ali, and U. Niesen, "Online coded caching," *IEEE/ACM Transactions on Networking*, vol. 24, no. 2, pp. 836–845, 2016.

[29] A. Shabani, S. P. Shariatpanahi, V. Shah-Mansouri, and A. Khonsari, "Mobility increases throughput of wireless device-to-device networks with coded caching," in *2016 IEEE International Conference on Communications (ICC)*, 2016, pp. 1–6.

[30] C. Shangguan, Y. Zhang, and G. Ge, "Centralized coded caching schemes: A hypergraph theoretical approach," *IEEE Transactions on Information Theory*, vol. 64, no. 8, pp. 5755–5766, 2018.

[31] K. Shanmugam, A. G. Dimakis, and M. Langberg, "Local graph coloring and index coding," *2013 IEEE International Symposium on Information Theory*, pp. 1152–1156, 2013.

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

[33] K. Shanmugam, A. M. Tulino, and A. G. Dimakis, "Coded caching with linear subpacketization is possible using ruzsa-szeméredi graphs," in *2017 IEEE International Symposium on Information Theory, ISIT 2017, Aachen, Germany, June 25-30, 2017*, 2017, pp. 1237–1241.

[34] S. P. Shariatpanahi, S. A. Motahari, and B. H. Khalaj, "Multi-server coded caching," *IEEE Transactions on Information Theory*, vol. 62, no. 12, pp. 7253–7271, 2016.

[35] L. Tang and A. Ramamoorthy, "Low subpacketization schemes for coded caching," in *2017 IEEE International Symposium on Information Theory (ISIT)*, 2017, pp. 2790–2794.

[36] ——, "Coded caching schemes with reduced subpacketization from linear block codes," *IEEE Transactions on Information Theory*, vol. 64, no. 4, pp. 3099–3120, 2018.

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

[38] ——, "An index coding approach to caching with uncoded cache placement," *IEEE Transactions on Information Theory*, vol. 66, no. 3, pp. 1318–1332, 2020.

[39] J. Wang, M. Cheng, Q. Yan, and X. Tang, "Placement delivery array design for coded caching scheme in d2d networks," *IEEE Transactions on Communications*, vol. 67, no. 5, pp. 3388–3395, 2019.

[40] Y.-P. Wei and S. Ulukus, "Coded caching with multiple file requests," in *2017 55th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, 2017, pp. 437–442.

[41] N. Woolsey, R.-R. Chen, and M. Ji, "Towards finite file packetizations in wireless device-to-device caching networks," *IEEE Transactions on Communications*, vol. 68, no. 9, pp. 5283–5298, 2020.

[42] 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.

[43] X. Wu, M. Cheng, C. Li, and L. Chen, "Design of coded caching schemes through proper orthogonal arrays," in *2022 IEEE International Symposium on Information Theory (ISIT)*, 2022, pp. 1306–1311.

[44] Q. Yan, M. Cheng, X. Tang, and Q. Chen, "On the placement delivery array design for centralized coded caching scheme," *IEEE Transactions on Information Theory*, vol. 63, no. 9, pp. 5821–5833, 2017.

[45] Q. Yan, U. Parampalli, X. Tang, and Q. Chen, "Online coded caching with random access," *IEEE Communications Letters*, vol. 21, no. 3, pp. 552–555, 2017.

[46] Q. Yan, X. Tang, Q. Chen, and M. Cheng, "Placement delivery array design through strong edge coloring of bipartite graphs," *IEEE Communications Letters*, vol. 22, no. 2, pp. 236–239, 2018.

[47] Q. Yan, M. Wigger, S. Yang, and X. Tang, "A fundamental storage-communication tradeoff for distributed computing with straggling nodes," *IEEE Transactions on Communications*, vol. 68, no. 12, pp. 7311–7327, 2020.

[48] ǫ. Yapar, K. Wan, R. F. Schaefer, and G. Caire, "On the optimality of d2d coded caching with uncoded cache placement and one-shot delivery," *IEEE Transactions on Communications*, vol. 67, no. 12, pp. 8179–8192, 2019.

[49] J. Zhang, X. Lin, and X. Wang, "Coded caching under arbitrary popularity distributions," *IEEE Transactions on Information Theory*, vol. 64, no. 1, pp. 349–366, 2018.

[50] X. Zhang, X. T. Yang, and M. Ji, "A new design framework on d2d coded caching with optimal rate and less subpacketizations," in *2020 IEEE International Symposium on Information Theory (ISIT)*, 2020, pp. 1699–1704.

[51] 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.

[52] ——, "Coded caching schemes with linear subpacketizations," *IEEE Transactions on Communications*, vol. 69, no. 6, pp. 3628–3637, 2021.

Min Xu received the B.S. degree in mathematics from Wuhan University, Wuhan, Hubei, China, in 2014. He is currently pursuing the Ph.D. degree with Nankai University, Tianjin, China. His research interests include combinatorics and network information theory.

Zixiang Xu received his B.S. degree in mathematics from Zhejiang University, Hangzhou, Zhejiang, P. R. China, in 2016, and received his Ph.D. degree in mathematics from Capital Normal University, Beijing, P. R. China, in 2022. He is currently a postdoctoral fellow at Extremal Combinatorics and Probability Group of Institute for Basic Science, Daejeon, South Korea. His research interests include extremal combinatorics, coding theory and discrete geometry.

Gennian Ge received the M.S. and Ph.D. degrees in mathematics from Suzhou University, Suzhou, Jiangsu, P. R. China, in 1993 and 1996, respectively. After that, he became a member of Suzhou University. He was a postdoctoral fellow in the Department of Computer Science at Concordia University, Montreal, QC, Canada, from September 2001 to August 2002, and a visiting assistant professor in the Department of Computer Science at the University of Vermont, Burlington, Vermont, USA, from September 2002 to February 2004. He was a full professor in the Department of Mathematics at Zhejiang University, Hangzhou, Zhejiang, P. R. China, from March 2004 to February 2013. Currently, he is a full professor in the School of Mathematical Sciences at Capital Normal University, Beijing, P. R. China. His research interests include combinatorics, coding theory, information security, and their interactions.

Dr. Ge is on the Editorial Board of *Journal of Combinatorial Theory, Series A, IEEE Transactions on Information Theory, Designs Codes and Cryptography, Journal of Combinatorial Designs, Journal of Algebraic Combinatorics, Science China Mathematics, Applied Mathematics−A Journal of Chinese Universities*. He received the 2006 Hall Medal from the Institute of Combinatorics and its Applications.

Min-Qian Liu received the B.S. and Ph.D. degrees in Statistics from Nankai University, Tianjin, P. R. China, in 1993 and 1999, respectively. He was an assistant professor from July 1998 to June 2001 and then an associate professor from June 2001 to August 2002 in the School of Management at Tianjin University, Tianjin, P. R. China. He was an associate professor in the Department of Statistics at Nankai University from August 2002 to December 2004, and a full professor at Nankai University since December 2004. Currently, he is a distinguished professor in the School of Statistics and Data Science at Nankai University. His research interests include design and analysis of experiments, design and modeling of computer experiments, and big data analysis.

Dr. Liu is on the Editorial Boards of *Communications in Statistics – Theory and Methods, Communications in Statistics – Simulation and Computation, Journal of Statistical Theory and Practice, Journal of Applied Statistics and Management, Statistics and Decision, and Journal of Statistics*.