Computer Networks and Communications
http://ojs.wiserpub.com/index.php/CNC/

UNIVERSAL WISER PUBLISHER

Research Article

Combinatorial Multi-Access Coded Caching with Private Caches under Intersecting Index Constraints

Dhruv Pratap Singh[1], Anjana A. Mahesh[2,*], B. Sundar Rajan[1]

[1] Department of Electrical Communication Engineering, Indian Institute of Science, Bangalore 560012, India
[2] Department of Electrical Engineering, Indian Institute of Technology Hyderabad, Telangana 502285, India
[*] Correspondence: anjana.am@ee.iith.ac.in

Received: 23 April 2026; **Revised:** 11 June 2026; **Accepted:** 24 June 2026; **Published:** 29 June 2026

Abstract: Modern cache-aided networks often involve users accessing multiple shared caches while possessing heterogeneous local storage resources. We consider combinatorial multi-access coded caching systems with private caches, where each user accesses multiple access caches and maintains a private memory. We study the impact of variable private cache sizes on coded multicast opportunities. We identify a structural regime, termed the intersection class, under which non-trivial coded caching is feasible, and derive a necessary and sufficient condition for this regime. For systems in this class, we characterize an achievable rate-memory trade-off. We further identify a subclass, referred to as the uniform-intersection subclass, for which the proposed scheme admits a regular transmission structure, wherein each transmission benefits the same number of users. Using an index coding formulation, we establish a converse bound and show that the proposed scheme operates close to optimality.

***Keywords*:** coded caching, combinatorial multi-access networks, private caches, intersection class, index coding, rate-memory trade-off, uncoded placement, cache-aided networks

1. Introduction

The rapid growth of bandwidth-intensive applications has placed significant strain on modern communication networks. To address this challenge, Maddah-Ali and Niesen (MAN) introduced coded caching in [1], in which the server prefetches file contents into user caches during off-peak hours and generates coded multicast transmissions during peak hours to reduce the delivery rate. This system operates in two phases: a *placement phase*, during which the network load is low, and caches are populated without knowledge of future demands, and a *delivery phase*, in which user requests are satisfied through coded transmissions. The delivery rate is defined as the number of transmitted bits normalized by the file size.

The original coded caching framework considers a set of users connected to a central server via an error-free wireless broadcast link, where each user is equipped with a dedicated cache. Under uncoded placement, the optimality of this framework has been established in [2, 3]. Several extensions have since been proposed to capture more general network architectures and operating assumptions, including decentralized and secretive coded caching for shared caches [4, 5], online coded caching [6], combinatorial multi-access coded caching (CMACC) networks [7, 8], hierarchical architectures [9], systems with offline users [10], satellite networks [11], two-dimensional networks [12], nonuniform demands [13],

Copyright ©2026 Dhruv Pratap Singh, et al.
DOI: https://doi.org/10.37256/cnc.41202610059
This is an open-access article distributed under a CC BY license
(Creative Commons Attribution 4.0 International License)
https://creativecommons.org/licenses/by/4.0/

demand-private caching [14], location-based systems [15], and heterogeneous user profiles [16], among others. More recently, the Combinatorial Multi-Access coded caching framework with Private caches (CMAP) was introduced in [17] to model systems in which users access multiple access caches and are additionally equipped with a private cache.

While [17] considers a fixed private cache size, the impact of variable private cache sizes on coded multicast opportunities has not been characterized. In practical systems, user devices may have heterogeneous and flexible storage capabilities, and understanding the role of the memory of the private caches is therefore of fundamental interest.

In this work, we study combinatorial multi-access coded caching systems with private caches and consider multiple values of the private cache memory subject to structural constraints. We define a class of system parameters, termed the intersection class, which specifies the admissible private cache memory values. We then derive a necessary and sufficient condition under which a given system belongs to this class. For systems in the intersection class, we propose a centralized coded caching scheme and characterize its achievable rate-memory trade-off. We further identify a uniform-intersection subclass of the intersection class, for which we derive the corresponding membership condition and show that the proposed scheme admits a regular delivery structure, wherein each transmission benefits an equal number of users; the associated rate-memory trade-off is also characterized. Finally, based on the placement policy of the proposed scheme, we derive an index-coding-based converse bound on the optimal worst-case delivery rate and demonstrate near-optimal performance.

Our Contributions: In this work, we introduce a class of system parameters, termed the intersection class, which specifies the private cache memory values admissible under structural constraints. For systems in this class, we propose a centralized coded caching scheme and characterize its achievable rate-memory trade-off. We further identify a uniform-intersection subclass of the intersection class and show that, within this subclass, the proposed scheme is regular, in the sense that each coded transmission benefits the same number of users; the corresponding rate-memory trade-off is also characterized. To establish fundamental limits, we derive an index coding-based lower bound on the minimum achievable delivery rate under the uncoded cache placement policy of the proposed scheme. Finally, we present numerical results comparing the proposed scheme with the derived lower bound as well as existing bounds reported in [17].

Organization of the paper: Section 2 introduces the system model along with the necessary preliminaries. In Section 3, we present the main results of this work. The proposed delivery scheme is described in Section 4, and Section 5 provides numerical comparisons with existing bounds from [17] and the lower bound proposed in Section 3.

Notations: The set $\{1,2,\cdots,N\}$ is denoted as $[N]$, whereas the set $\{a,a+1,\cdots,b\}$ where $b \geq a, a \neq 1$ is denoted by $[a,b]$, for some $a,b \in \mathbb{Z}^+$, where $\mathbb{Z}^+$ is the set of all non-negative integers. The cardinality of a set A is denoted as $|A|$. We use $A_{i,i\in[n]}$ to denote $\{A_1,A_2,\cdots,A_n\}$. The binomial coefficient $\frac{n!}{k!(n-k)!}$ is denoted as $\binom{n}{k}$ and we assume $\binom{n}{k}=0$ if $n<0, k<0$ or $n<k$. We use the $\oplus$ symbol to denote the bitwise XOR operation.

2. System model and preliminaries

In this section, we introduce the system model. We then revisit relevant bounds from [17] and results from index coding that will be used in this work.

2.1 *System model*

Consider the system model shown in Figure 1. A central server stores a library of N files, $W_1,W_2,\cdots,W_N$, each of size B bits. There are Λ caches in the system, referred to as access caches, each with a storage capacity of M_aB bits, where $M_a \leq N$. In addition, there are K users, each equipped with a private cache of size $M_p \leq N$, that connect to the access caches through error-free wireless links of infinite capacity. The server communicates with all users through a shared, error-free broadcast link.

Each user connects to a distinct r-subset of the Λ access caches, where r denotes the access degree. Accordingly, the users are indexed by subsets of $[\Lambda]$ of cardinality r. This combinatorial multi-access topology [7, 8], results in $K=\binom{\Lambda}{r}$ users, where $K \leq N$.

We focus on the memory points (M_a,M_p) satisfying $0 < rM_a + M_p < N$. The resulting system is referred to as the (Λ,r,M_a,M_p,N)-CMAP coded caching system. The system operates in two phases:

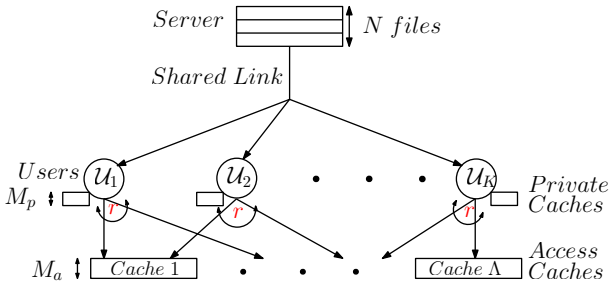


Figure 1. The $(\Lambda, r, M_a, M_p, N)$-CMAP coded caching system

Placement Phase: The server populates both the private and access caches using an uncoded placement strategy. Define $t_a = \frac{\Lambda M_a}{N} \in \mathbb{Z}^+$ and $t_p = \frac{KM_p}{N} \in \mathbb{Z}^+$. Each file $W_n, n \in [N]$ is first divided into subfiles as $W_n = \{W_{n,S} : S \subset [\Lambda], |S| = t_a\}$, which are stored in the access caches such that cache i stores the subfiles $W_{n,S}$, for all $n \in [N]$, if $i \in S$. Next, each subfile $W_{n,S}$ is further partitioned into mini-subfiles as $W_{n,S} = \{W_{n,S,T_{i,i\in[t_p]}} : T_i \subset [\Lambda] \setminus S, |T_i| = r, \forall i \in [t_p]\}$. Each mini-subfile $W_{n,S,T_{i,i\in[t_p]}}$ is prefetched in the private caches of the users $T_1, T_2, \cdots, T_{t_p}$.

The total number of mini-subfiles each file is divided into is referred to as the subpacketization level and is denoted by F. The contents stored in access cache $i \in [\Lambda]$ and the private cache of user $U \subseteq [\Lambda], |U| = r$ are denoted by Z_i and Z_U^p, respectively.

Delivery Phase: Each user $U \subseteq [\Lambda], |U| = r$ requests one file from the server. The collection of all user demands is represented by the demand vector $\mathbf{d} = (d_U : U \subseteq [\Lambda], |U| = r)$, where $d_U \in [N]$ denotes the index of the file requested by user U.

We denote $W_{d_U,S,T_{i,i\in[t_p]}}$ as the mini-subfile corresponding to subfile S, which is stored in the private cache of users $T_1, T_2, \cdots, T_{t_p}$ and is demanded by user U. Upon receiving the demand vector $\mathbf{d}$, the server constructs coded multicast transmissions by combining appropriately chosen mini-subfiles so that each transmission is simultaneously useful to multiple users. Using the transmissions along with the contents of their private and accessed caches, each user decodes their requested file. The objective is to minimize the number of bits transmitted by the server.

The rate of the scheme is defined as the total number of bits transmitted by the server, normalized by the file size B. The worst-case rate corresponds to the scenario where all users demand distinct files.

Definition 1 Consider a $(\Lambda, r, M_a, M_p, N)$-CMAP coded caching system. A rate-memory triplet (M_a, M_p, R) is said to be achievable if there exists a coded caching scheme, under any general placement, that achieves rate R at memory point (M_a, M_p), for sufficiently large file size B, for any demand vector. The optimal worst-case rate for the $(\Lambda, r, M_a, M_p, N)$-CMAP system is defined as

$$R^*_{M_a,M_p} = \inf\{R : (M_a, M_p, R) \text{ is achievable}\}, \tag{1}$$

The server seeks to design jointly the placement and delivery policies to achieve $R^*_{M_a,M_p}$.

2.2 *Some bounds discussed in [17]*

In this subsection, we summarize some known bounds from [17] that will be used for comparisons in Section 5.

Theorem 1 For a $(\Lambda, r, M_a, M_p, N)$-CMAP coded caching system, the optimal worst-case rate under uncoded placement, denoted by $R^*_{UC}(M_a, M_p)$, is bounded as

$$R^*_D(rM_a + M_p) \leq R^*_{UC}(M_a, M_p) \leq R^*_{CMACC}(M_a + \tfrac{M_p}{r}),$$

where $R_D^*(rM_a+M_p)$ is the rate achieved by MAN scheme [1] with cache size $M = rM_a+M_p$ and $R^*_{CMACC}(M_a+\frac{M_p}{r})$ is the rate achieved by MAN scheme for CMACC network [8] with cache size $M = M_a+\frac{M_p}{r}$.

Theorem 1 bounds the optimal worst-case rate under uncoded placement for the CMAP system by relating it to the corresponding optimal rates under uncoded placement of the MAN scheme [1] and the MAN scheme for the CMACC system [8].

Theorem 2 For a $(\Lambda, r, M_a, M_p, N)$-CMAP coded caching system, the worst-case rate is lower bounded as

$$R^*(M_a, M_p) \geq \max_{s\in\{1,2,\cdots,K\}} \left(s - \frac{qM_a+sM_p}{\left\lfloor \frac{N}{s} \right\rfloor} \right),$$

where $q = \min\{\Lambda+r-1, \Lambda\}$.

In contrast to Theorem 1, Theorem 2 provides a general lower bound on the optimal worst-case rate for the CMAP system under arbitrary placement based on a cut-set-based converse argument.

2.3 *Index coding*

The Index Coding Problem (ICP) with side information [18], [19] involves a single source that has n messages $x_1, x_2, \ldots, x_n$, where $x_i \in \mathbb{F}_q$ for all $i \in [n]$, and broadcasts to a set of K receivers $R_1, R_2, \ldots, R_K$. Each receiver $R_i : i \in [K]$, has prior knowledge of a subset $\{x_j : j \in X_i\}$, where $X_i \subseteq [n]$ denotes its side information. Further, receiver R_i requests a single message $x_{f(i)}$, where $f : [K] \mapsto [n]$ and $f(i) \notin X_i$.

For an instance $\mathscr{I}$ of the ICP, the generalized independence number $\alpha(\mathscr{I})$ was defined in [20] as follows. For each receiver R_i, let $\mathscr{Y}_i = [n] \setminus (\{f(i)\} \cup X_i)$ and define $\mathscr{J}(\mathscr{I}) = \bigcup_{i\in[K]} \{\{f(i)\} \cup Y_i : Y_i \subseteq \mathscr{Y}_i\}$. A subset $H \subseteq [n]$ is called a generalized independent set if every subset of H belongs to $\mathscr{J}(\mathscr{I})$. The largest such set is referred to as the maximal generalized independent set, and its cardinality, denoted by $\alpha(\mathscr{I})$, is called the generalized independence number. It was shown in [21] that $\alpha(\mathscr{I})$ provides a lower bound on the number of scalar linear transmissions required to solve the ICP $\mathscr{I}$.

In the coded caching setting, for a given uncoded placement scheme and demand vector, each user can be modeled as a receiver in an ICP. The desired message corresponds to the subfiles of the requested file, while the side information consists of the subfiles available in the user's cache. As a result, the delivery phase of the coded caching problem can be formulated as an instance of the ICP, and the corresponding generalized independence number provides a lower bound on the number of scalar linear transmissions required to satisfy all user demands [21].

3. Main results

In this section, we define the intersection class for the CMAP coded caching model and characterize the system parameters for which a $(\Lambda, r, M_a, M_p, N)$-CMAP coded caching system belongs to the intersection class. For systems in the intersection class, we establish the achievable rate-memory trade-off of the proposed scheme. We further define the uniform-intersection subclass and derive the conditions under which a system belongs to this subclass. For systems in the uniform-intersection subclass, we show that the proposed scheme is regular, in the sense that each coded transmission serves the same number of users, and characterize the corresponding rate-memory trade-off. Finally, we derive an index coding-based lower bound on the optimal worst-case delivery rate under the uncoded placement strategy of the proposed scheme.

Definition 2 A $(\Lambda, r, M_a, M_p, N)$-CMAP coded caching system is said to belong to the intersection class if, for every mini-subfile, the index set of the requesting user has a non-empty intersection with the intersection of the corresponding private-cache index sets. More precisely, for any mini-subfile $W_{d_U, S, T_{i,i\in[t_p]}}$, the system belongs to the intersection class if the intersection set $I = U \cap \bigcap_{i=1}^{t_p} T_i \neq \emptyset$, where $t_p = \frac{KM_p}{N}$.

Intuitively, the intersection condition requires that, for every mini-subfile, there exists at least one access cache that is simultaneously connected to the requesting user and to all users that have prefetched that mini-subfile in their private caches. This imposes a combinatorial constraint on the system parameters Λ, r, M_a, M_p, and N, since they determine how the index sets U and $T_i, i \in [t_p]$ overlap. The following claim characterizes when this condition is satisfied.

Claim 1 A $(\Lambda, r, M_a, M_p, N)$-CMAP coded caching system belongs to the intersection class if and only if

$$\Lambda < t_a + r + \tfrac{r}{t_p},$$

where $t_a = \frac{\Lambda M_a}{N}$ and $t_p = \frac{KM_p}{N}$.

Proof. We prove this claim in two steps. First, we prove that for $\Lambda < t_a + r + \frac{r}{t_p}$, there exists no mini-subfile such that the intersection set $I = \emptyset$. This is proven via contradiction. Following this, we show that for any $\Lambda \geq t_a + r + \frac{r}{t_p}$, mini-subfiles with $I = \emptyset$ will always exist. We now proceed with the first part of the proof.

1) *Proof by Contradiction*: Assume that for $\Lambda < t_a + r + \frac{r}{t_p}$, there exists a user-index set U and mini-subfile-index sets $T_{i,i\in[t_p]}$ such that their intersection is empty, i.e., the intersection set $I = \{\bigcap_{i=1}^{t_p} T_i\} \cap U = \emptyset$. Each set U and $T_{i,i\in[t_p]}$ has cardinality r and is a subset of $[\Lambda] \setminus S$, where $S \subset [\Lambda]$ and $|S| = t_a$.

For each integer $j \in [\Lambda] \setminus S$ appearing in these sets, let i_j denote its number of occurrences across the $(t_p + 1)$ sets, U and $T_{i,i\in[t_p]}$. It follows that $i_j \leq t_p$ because if $i_j = t_p + 1$, then $I \neq \emptyset$, contradicting our assumption. Given that $j \in [\Lambda] \setminus S$, we have

$$\sum_{j\in[\Lambda]\setminus S} i_j \leq (\Lambda - t_a)t_p.$$

But from the claim, we have $(\Lambda - t_a)t_p < (t_p + 1)r$, leading to

$$\sum_{j\in[\Lambda]\setminus S} i_j < (t_p + 1)r.$$

This is a contradiction as each set has cardinality r, so the sum of occurrences of all the integers $\sum_{j\in[\Lambda]\setminus S} i_j = (t_p + 1)r$. Hence, for $\Lambda < t_a + r + \frac{r}{t_p}$, there exists no mini-subfile such that the intersection set $I = \emptyset$.

We now present the second part of the proof.

2) *Counter-example for* $\Lambda \geq t_a + r + \frac{r}{t_p}$: Having established that $I \neq \emptyset$ for $\Lambda < t_a + r + \frac{r}{t_p}$, we now demonstrate that for any $\Lambda \geq t_a + r + \frac{r}{t_p}$, there always exist mini-subfiles such that $I = \emptyset$.

Consider $\Lambda = t_a + r + \frac{r}{t_p}$. Since Λ, t_a and r are integers, it follows that t_p divides r.

Partition the set $[\Lambda] \setminus S$ into $(t_p + 1)$ subsets $P_{i,i\in[t_p+1]}$ of cardinality $\frac{\Lambda - t_a}{t_p+1} = \frac{r}{t_p}$. For each P_i, define $Q_i = [\Lambda] \setminus \{S \cup P_i\}$. Note that $|Q_i| = \Lambda - t_a - \frac{\Lambda - t_a}{t_p+1} = \frac{(\Lambda - t_a)t_p}{t_p+1} = r$. Since the sets $Q_{i,i\in[t_p+1]}$ are sets of cardinality r, they serve as placeholders for the user-index set U and the mini-subfile-index sets $T_{i,i\in[t_p]}$.

Consider an element $e \in [\Lambda] \setminus S$. Since the sets $P_{i,i\in[t_p+1]}$ partition the set $[\Lambda] \setminus S$, we know that e belongs in one P_i, that is, $e \in P_i$ for some i, say i^*. Thus, we have $e \in \{Q_i : \forall i \in [t_p + 1] \setminus \{i^*\}\}$. Hence, there exists Q_{i^*} such that $e \notin Q_{i^*}$. Therefore, we have $\bigcap_{i=1}^{t_p+1} Q_i = \emptyset$.

Thus, there will always exist a mini-subfile $W_{d_U, S, T_{i,i\in[t_p]}}$ such that $I = \{\bigcap_{i=1}^{t_p} T_i\} \cap U = \emptyset$. □

The condition $\Lambda < r + t_a + \frac{r}{t_p}$ ensures that the available access cache indices are sufficiently crowded so that the requesting user and the users storing a given mini-subfile must share at least one common cache. When the total number of available cache indices Λ is small relative to the parameters r, t_a, and t_p, it becomes impossible to assign disjoint index sets

without forcing an overlap. In the absence of such an overlap, the proposed coded transmissions cannot be formed, and the delivery reduces to essentially uncoded transmissions.

Theorem 3 For a $(\Lambda, r, M_a, M_p, N)$-CMAP coded caching system such that $\Lambda < t_a + r + \frac{r}{t_p}$, a worst-case rate

$$R = \sum_{i=1}^{r-1} \frac{\binom{\Lambda-t_a}{i}\sum_{j=0}^{r-i-1}(-1)^j\binom{\Lambda-t_a-i}{j}\binom{\binom{\Lambda-t_a-i-j}{r-i-j}}{t_p+1}}{\binom{\binom{\Lambda-t_a}{r}}{t_p}\binom{t_a+i}{t_a}}, \tag{2}$$

is achievable for the subpacketization $F = \binom{\Lambda}{t_a}\binom{\binom{\Lambda-t_a}{r}}{t_p}$, where $t_a = \frac{\Lambda M_a}{N} \in [0,\Lambda]$ and $t_p = \frac{KM_p}{N} \in [0,K]$.

Proof. Section 4.1 presents the coded caching scheme that achieves the rate stated in Theorem 3. □

The achievable rate in the above theorem is obtained by counting the number of demanded mini-subfiles and the number of users simultaneously served by each coded transmission under the proposed placement policy.

Remark 1 For $M_p = \frac{N}{K}$, the above expression reduces to the corresponding rate in [17], thereby recovering the previously studied setting. Moreover, within the intersection class, increasing M_p enlarges the effective multicasting gain through additional private-cache side information.

Remark 2 Note that while the rate in Theorem 3 is defined for $t_a, t_p \in \mathbb{Z}^+$, the lower convex envelope of the points in Theorem 3 is achievable via memory sharing.

We now define the uniform-intersection subclass.

Definition 3 A $(\Lambda, r, M_a, M_p, N)$-CMAP coded caching system belongs to the uniform-intersection subclass if the cardinality of the intersection set I is constant across all mini-subfiles.

Note that the uniform-intersection property requires that each mini-subfile be associated with the same number of common access caches across all users. This imposes additional symmetry constraints on the placement structure, which translate into specific combinatorial conditions on the system parameters. The following claim characterizes when these conditions are satisfied.

Claim 2 A $(\Lambda, r, M_a, M_p, N)$-CMAP coded caching system belongs to the uniform-intersection subclass if the following two conditions hold:

1) $\binom{r}{i}\binom{\binom{\Lambda-t_a-i}{r-i}-1}{t_p} = \binom{\binom{\Lambda-t_a}{r}-1}{t_p}$,

2) $\binom{\Lambda-t_a-i-1}{r-i-1} < t_p + 1 \leq \binom{\Lambda-t_a-i}{r-i}$, where $|I| = i$, $i \in [1, r-1]$, $t_a = \frac{\Lambda M_a}{N} \in [0,\Lambda]$, and $t_p = \frac{KM_p}{N} \in [0,K]$.

Proof. The proof of Claim 2 is provided in Section 4.2. The proof is based on a combinatorial counting argument. For each integer k, we first characterize the number of demanded mini-subfiles whose intersection sets have cardinality at least k, and then use an inclusion-exclusion argument to determine the number of mini-subfiles whose intersection sets have cardinality exactly k. Since, in the uniform-intersection subclass, all mini-subfiles have the same intersection size $|I|$, it follows that the number of mini-subfiles with larger intersection cardinality must be zero. This observation yields the second condition. Next, we compute the total number of demanded mini-subfiles in two different ways: first, by summing the contributions corresponding to intersection size $|I|$, and second, by direct combinatorial enumeration based on the placement policy. Equating these two expressions and simplifying using standard binomial identities leads to the first condition. Combining the two parts establishes the claim. □

Corollary 1 For a $(\Lambda, r, M_a, M_p, N)$-CMAP coded caching system satisfying the conditions in Claim 2, the worst-case rate

$$R = \frac{\binom{\Lambda-t_a}{r}-t_p}{(t_p+1)\binom{t_a+i}{t_a}},$$

is achievable, with a subpacketization level $F = \binom{\Lambda}{t_a}\binom{\binom{\Lambda-t_a}{r}}{t_p}$, where $t_a = \frac{\Lambda M_a}{N} \in [0,\Lambda]$ and $t_p = \frac{KM_p}{N} \in [0,K]$.

Proof. Under the conditions in Claim 2, all demanded mini-subfiles have the same intersection cardinality $|I| = i$. As a result, each coded transmission serves the same number of users.

The number of transmissions is obtained by dividing the total number of demanded mini-subfiles by the number of users served by each coded transmission. Normalizing this quantity by the subpacketization gives the achievable rate. Substituting these expressions and simplifying using binomial identities yields the stated rate expression. The detailed derivation is provided in Section 4.2. □

Theorem 4 (Alpha Bound) For a $(\Lambda, r, M_a, M_p, N)$-CMAP coded caching system and the placement policy described in Section 4.1, the optimal worst-case rate under uncoded placement is lower bounded as

$$R^*_{UC}(M_a, M_p) \geq \frac{\sum_{i=1}^{\Lambda - t_a - r + 1} \binom{\Lambda - r - i + 1}{t_a} \left(\binom{\Lambda - t_a}{r \; t_p} - i \right) + \sum_{m=\Lambda - t_a - r + 2}^{\binom{\Lambda - t_a}{r}} \binom{\binom{\Lambda - t_a}{r} - m}{t_p}}{\binom{\Lambda}{t_a} \binom{\Lambda - t_a}{r \; t_p}}, \tag{3}$$

where $t_a = \frac{\Lambda M_a}{N} \in [0, \Lambda]$ and $t_p = \frac{K M_p}{N} \in [0, K]$.

Proof. The proof is presented in Section 4.3. □

This lower bound is obtained by modeling the delivery phase under the proposed placement as an instance of the index coding problem with side information. The bound reflects the minimum number of transmissions required to satisfy all users, given the side information induced by the cache content available to the users. While the lower bound presented in Theorem 4 holds for any $(\Lambda, r, M_a, M_p, N)$-CMAP coded caching system, we apply this lower bound as a baseline for evaluating the performance of the CMAP system considered in this work.

4. Achievability scheme and lower bound

In this section, we describe the placement and delivery scheme that achieves the rate characterized in Theorem 3 and present the proof of the index-coding based lower bound stated in Theorem 4.

4.1 *Achievability scheme*

Before presenting the delivery scheme, we introduce two functions, denoted as $flip$ and $swap_o$, which are used to construct the coded transmissions.

Definition 4 For the mini-subfile $W_{d_{U,S,T_{i,i\in[t_p]}}}$, the function $flip(W_{d_{U,S,T_{i,i\in[t_p]}}})$ is defined as $flip(W_{d_{U,S,T_{i,i\in[t_p]}}}) = W_{d_{U,S,T_{i,i\in[t_p]}}} \oplus \bigoplus_{j\in[t_p]} W_{d_{T_j,S,\{U,T_{i,i\in[t_p]}\setminus\{j\}\}}}$. Further, the $flip$ of a coded combination of mini-subfiles is defined as the coded combination of the $flip$ of the individual mini-subfiles as $flip\left(\bigoplus_{i=1}^{n} W_{d_{U^i,S^i,T^i_{j,j\in[t_p]}}}\right) = \bigoplus_{i=1}^{n} flip(W_{d_{U^i,S^i,T^i_{j,j\in[t_p]}}})$.

In other words, the $flip$ function exchanges the user-index set U with each of the mini-subfile-index sets $T_i, i \in [t_p]$ and forms the XOR of the resulting $t_p + 1$ mini-subfiles.

Definition 5 For the mini-subfile $W_{d_{U,S,T_{i,i\in[t_p]}}}$ such that the intersection set $I = U \cap \bigcap_{j\in[t_p]} T_j \neq \emptyset$, we define the function $swap_o(W_{d_{U,S,T_{j,j\in[t_p]}}}, i)$ as $swap_o(W_{d_{U,S,T_{j,j\in[t_p]}}}, i) = \bigoplus_{\widetilde{U}\subset I, |\widetilde{U}|=i, \widetilde{S}\subset S, |\widetilde{S}|=i} W_{d_{\{U\cup\widetilde{S}\}\setminus\widetilde{U},\{S\cup\widetilde{U}\}\setminus\widetilde{S},\{\{\{T_j\cup\widetilde{S}\}\setminus\widetilde{U}\}_{j\in[t_p]}\}}}$, where $i \in [\min\{t_a, |I|\}]$.

The function $swap_o$ exchanges an i-subset of the intersection set $I = U \cap \bigcap_{j\in[t_p]} T_j$ with an i-subset of the subfile-index set S, and forms the XOR of the resulting mini-subfiles.

These operators are used jointly to generate coded transmissions that simultaneously serve multiple users.

We now present an example that illustrates the proposed scheme and demonstrates the role of the functions defined above. Following the convention in [17], we write the user-index set U, subfile-index set S, and mini-subfile-index sets $T_{i,i\in[t_p]}$ without explicit set notation.

Example 1 Consider a CMAP coded caching system in which a central server stores a library of $N = 10$ files. The system has $\Lambda = 5$ access caches, each with a memory of $M_a = 2$ files. Users connect to distinct subsets of $r = 3$ access

caches, resulting in $K = \binom{\Lambda}{r} = 10$ users. Each user is also equipped with a private cache of size $M_p = 2$ files. The users are indexed by the subsets of access caches they connect to, namely, $123, 124, 125, 134, 135, 145, 234, 235, 245$, and 345.

For this setting, we have $t_a = \frac{\Lambda M_a}{N} = 1$. Each file is divided into $\binom{\Lambda}{t_a} = 5$ subfiles, indexed as $W_n = \{W_{n,i} : i \in [\Lambda]\}$. The contents of access cache i are given by $Z_i = \{W_{n,i}, \forall n \in [N]\}$. For example, $Z_1 = \{W_{n,1}, \forall n \in [10]\}$. Each access cache stores $\frac{10}{5} = 2$ files, satisfying its memory constraint.

Next, we have $t_p = \frac{KM_p}{N} = 2$ and the server breaks each subfile $W_{n,S}$ into $\binom{\binom{\Lambda - t_a}{r}}{t_p} = 6$ mini-subfiles as $W_{n,S} = \{W_{n,S,T_{i\in[t_p]}} : T_i \subset [\Lambda] \setminus S, |T_i| = r, \forall i \in [t_p]\}$. For example, the subfile $W_{n,1} = \{W_{n,1,234,235}, W_{n,1,234,245}, W_{n,1,234,345}, W_{n,1,235,245}, W_{n,1,235,345}, W_{n,1,245,345}\}$. The private caches are populated with the mini-subfiles corresponding to the subfiles not available to the user from the access caches. The contents of the private cache Z_U^p are given as $Z_U^p = \{W_{n,S,T_{i,i\in[t_p]}} : S \subseteq [\Lambda] \setminus U, T_1 = U, \{T_i \in \{T' \subseteq [\Lambda] \setminus S, |T'| = r\}, \forall i \in [2, t_p]\}, \forall n \in [N]\}$. For example, $Z_{123}^p = \{W_{n,4,123,125}, W_{n,4,123,135}, W_{n,4,123,235}, W_{n,5,123,124}, W_{n,5,123,134}, W_{n,5,123,234}, \forall n \in [10]\}$. Each private cache stores $\frac{6\times 10}{6\times 5} = 2$ files, satisfying its memory constraint.

Finally, as $t_a + r + \frac{r}{t_p} = 1 + 3 + \frac{3}{2} = 5.5 > \Lambda = 5$, this system falls in the intersection class.

We now explain how the transmissions in the delivery scheme are constructed. For a demand vector $\mathbf{d} = (d_{123}, d_{124}, d_{125}, d_{134}, d_{135}, d_{145}, d_{234}, d_{235}, d_{245}, d_{345})$, consider a mini-subfile $W_{d_{123},4,125,135}$ demanded by the user 123.

For this mini-subfile, the intersection set $I = \{123 \cap 125 \cap 135\} = \{1\}$. The server first applies the function $swap_o(W_{d_{123},4,125,135}, 1)$ which generates $W_{d_{234},1,245,345}$. Next, the server employs the function $flip(W_{d_{123},4,125,135} \oplus W_{d_{234},1,245,345})$, producing the transmission $W_{d_{123},4,125,135} \oplus W_{d_{125},4,123,135} \oplus W_{d_{135},4,123,125} \oplus W_{d_{234},1,245,345} \oplus W_{d_{245},1,234,345} \oplus W_{d_{345},1,234,245}$.

The mini-subfiles $W_{d_{234},1,245,345}, W_{d_{245},1,234,345}$, and $W_{d_{345},1,234,245}$ are present with the user 123 via Z_1. Further, the mini-subfiles $W_{d_{125},4,123,135}$ and $W_{d_{135},4,123,125}$ are stored in its private cache. Hence, the user is able to decode the transmission. A similar argument holds for the other users in the above transmission. The remaining transmissions are constructed similarly by applying the operators $swap_o$ and $flip$ to the remaining demanded mini-subfiles. In total, the server generates 10 transmissions. Since the subpacketization level is $F = 5 \times 6 = 30$, the resulting rate is $R = \frac{10}{30} = 0.33$

Remark 3 In the above example, Claim 2 is satisfied, ensuring that the intersection set of each mini-subfile has cardinality one. As a result, the proposed scheme has a regular structure in which each transmission benefits the same number of users.

A total of 60 mini-subfiles are demanded, and by Theorem 4, the server must transmit at least 7 transmissions. In contrast, any regular scheme would require at least 10 transmissions, since 10 is the smallest integer greater than 7 that divides 60.

This demonstrates that the scheme in Example 1 is optimal under uncoded placement and regular transmission.

We now give the general description of the placement and delivery phase of the proposed scheme.

Placement Phase: Each file is first divided into $\binom{\Lambda}{t_a}$ non-overlapping subfiles of equal size as $W_n = \{W_{n,S} : S \subseteq [\Lambda], |S| = t_a\}$, where $t_a = \frac{\Lambda M_a}{N}$ denotes the access cache replication factor. The contents of access cache i are given by $Z_i = \{W_{n,S} : i \in S, S \subseteq [\Lambda], |S| = t_a, \forall n \in [N]\}$ for $i \in [\Lambda]$. Each access cache stores $\frac{N\binom{\Lambda-1}{t_a-1}}{\binom{\Lambda}{t_a}} = M_a$ files, satisfying its memory constraint.

We now describe how the private caches are populated. Each subfile is further split into $\binom{\binom{\Lambda - t_a}{r}}{t_p}$ mini-subfiles as $W_{n,S} = \{W_{n,S,T_{i,i\in[t_p]}} : T_i \subset [\Lambda] \setminus S, |T_i| = r, \forall i \in [t_p]\}$, where $t_p = \frac{KM_p}{N}$ is the private cache memory replication factor.

We denote the mini-subfile of subfile S of the file n as $W_{n,S,T_{i,i\in[t_p]}}$ where $T_i, i \in [t_p]$ are the users possessing the mini-subfile $W_{n,S,T_{i,i\in[t_p]}}$ in their private cache. The contents of the private cache of user U are $Z_U^p = \{W_{n,S,T_{i,i\in[t_p]}} : S \subseteq [\Lambda] \setminus U, T_1 = U, \{T_i \in \{T' \subseteq [\Lambda] \setminus S, |T'| = r\}, \forall i \in [2, t_p]\}, \forall n \in [N]\}$. The private cache of each user stores $\frac{N\binom{\Lambda-r}{t_a}\left(\binom{\Lambda-t_a}{r}-1\right)}{\binom{\Lambda}{t_a}\left(\binom{\Lambda-ta}{r}\right)} = M_p$ files, satisfying its memory constraint.

As per the proposed placement policy, there is no overlap between the contents of a user's private cache and the contents of the access caches the user connects to. Notably, for $t_p = 1$, this placement specializes to the one discussed in [17].

Delivery Phase: For each user U, define the user-demand set D_U as $D_U = \{(S, T_{i,i\in[t_p]}) : S \subseteq [\Lambda], |S| = t_a, T_i \subseteq [\Lambda], |T_i| = r, T_i \neq U, \forall i \in [t_p]\}$. In other words, the user-demand set D_U contains the indices of all the mini-subfiles demanded by user U.

For every element $(S, T_{i,i\in[t_p]}) \in D_U$, the server transmits $flip\big(W_{d_U,S,T_{l,l\in[t_p]}} \bigoplus_{k\in[\min\{|I|,t_a\}]} swap_o\big(W_{d_U,S,T_{l,l\in[t_p]}}, k\big)\big)$, where $I = U \cap \bigcap_{i=1}^{t_p} T_i$.

After each transmission, the server removes the corresponding mini-subfile indices from the user-demand sets of all the involved users. The server continues generating transmissions until all user-demand sets are empty, that is, $D_U = \emptyset, \forall U \subset [\Lambda], |U| = r$.

Decodability: For an element $(S, T_{i,i\in[t_p]})$ in the user-demand set D_U of user U, consider the transmission $flip\big(W_{d_U,S,T_{l,l\in[t_p]}} \oplus \bigoplus_{k\in[\min\{|I|,t_a\}]} swap_o\big(W_{d_U,S,T_{l,l\in[t_p]}}, k\big)\big)$, where $I = U \cap \bigcap_{i=1}^{t_p} T_i$. This can be rewritten as $flip\big(W_{d_U,S,T_{l,l\in[t_p]}}\big) \oplus \bigoplus_{k\in[\min\{|I|,t_a\}]} flip\big(swap_o\big(W_{d_U,S,T_{l,l\in[t_p]}}, k\big)\big)$.

Now consider the mini-subfiles in the second term. Each is obtained by swapping k elements from the sets $I \subseteq U$ and S. Hence, the resulting subfile indices necessarily intersect with U. By the access cache placement policy, user U has access to all these mini-subfiles.

Next, consider the mini-subfiles in $flip(W_{d_U,S,T_{l,l\in[t_p]}})$. User U has all mini-subfiles except $W_{d_U,S,T_{l,l\in[t_p]}}$, since the remaining terms include U in their mini-subfile-index sets and are therefore stored in the user's private cache. Hence, user U can cancel all known mini-subfiles in the transmission and decode its desired mini-subfile.

This decoding process holds for all users involved in the transmission. Since the server continues transmitting until all user-demand sets are empty, all user demands are satisfied.

Rate Characterization: We first characterize the total number of demanded mini-subfiles $W_{d_U,S,T_{i,i\in[t_p]}}$ having an intersection set $I = U \cap \bigcap_{j\in[t_p]} T_j$ of cardinality i, where $i \in [1, r-1]$.

Consider a specific subfile-index set S of cardinality t_a and an intersection set I of cardinality i. The problem of determining the number of mini-subfiles of a given subfile S having an intersection set I of cardinality $|I| = i$ reduces to identifying (t_p+1) sets $A_{j,j\in[t_p+1]}$ of cardinality r such that their intersection satisfies $\bigcap_{j=1}^{t_p+1} A_j = I$. Once these sets are identified, one set is chosen as the user-index set U and the remaining sets form the mini-subfile-index sets $T_{j,j\in[t_p]}$. Moving forward, we denote the cardinality of the intersection I as $|I|$. We now outline the procedure for calculating the number of collections of such sets.

To obtain the number of collections of (t_p+1) sets $A_{j,j\in[t_p]}$ of cardinality r whose intersection is exactly equal to a given set I, we begin by counting the number of such collections where each set contains all elements of I. Out of the set $[\Lambda]$, we remove t_a elements for the subfile-index set S, and $|I|$ elements are reserved in every set A_j to ensure an intersection set of I. From the remaining elements, we pick $r-|I|$ elements to form candidate sets, from which we choose (t_p+1) sets, yielding $\binom{\binom{\Lambda-t_a-|I|}{r-|I|}}{t_p+1}$ such collections. To ensure that the intersection is exactly equal to I, we apply the Principle of Inclusion-Exclusion (PIE). This method alternately subtracts and adds the number of collections where the sets contain additional common elements, leading to $\sum_{j=0}^{r-|I|-1}(-1)^j\binom{\Lambda-t_a-|I|}{j}\binom{\binom{\Lambda-t_a-|I|-j}{r-|I|-j}}{t_p+1}$ collections of (t_p+1) sets having an intersection set I.

To determine the total number of mini-subfiles corresponding to these (t_p+1) sets, we designate one among them as the user-index set U, and the remaining t_p sets form the mini-subfile-index sets $T_{i,i\in[t_p]}$. Since the selection of a user-index set U can be made in (t_p+1) ways, the number of mini-subfiles of a particular subfile associated with the intersection set I is $(t_p+1)\sum_{j=0}^{r-|I|-1}(-1)^j\binom{\Lambda-t_a-|I|}{j}\binom{\binom{\Lambda-t_a-|I|-j}{r-|I|-j}}{t_p+1}$.

Finally, the total number of mini-subfiles corresponding to any subfile of cardinality t_a and an intersection set I of cardinality $|I|$ is $\binom{\Lambda}{t_a}(t_p+1)\binom{\Lambda-t_a}{|I|}\sum_{j=0}^{r-|I|-1}(-1)^j\binom{\Lambda-t_a-|I|}{j}\binom{\binom{\Lambda-t_a-|I|-j}{r-|I|-j}}{t_p+1}$. Here, the terms $\binom{\Lambda}{t_a}$ and $\binom{\Lambda-t_a}{|I|}$ account for the $\binom{\Lambda}{t_a}$ possible subfiles and $\binom{\Lambda-t_a}{|I|}$ possible subsets of cardinality $|I|$ of the set $[\Lambda] \setminus S$.

Having determined the number of mini-subfiles that yield an intersection set of cardinality $|I|$, we now calculate the number of such mini-subfiles included in a transmission.

For an intersection set of cardinality $|I|$, each transmission is a coded combination of $(t_p+1)\sum_{l=0}^{\min\{|I|,t_a\}}\binom{|I|}{l}\binom{t_a}{l}$ mini-subfiles. This is because the function $swap_o$ exchanges l elements of the intersection set $I=\{\bigcap_{k=1}^{t_p}T_k\}\cap U$ with l elements of the subfile-index set S, where $|S|=t_a$ and $l\in[min\{|I|,t_a\}]$. Further, the function $flip$ exchanges the user-index set with one of the mini-subfile-index sets, generating (t_p+1) mini-subfiles for each mini-subfile generated by the function $swap_o$. Using the Vandermonde's identity, we simplify $(t_p+1)\sum_{l=0}^{\min\{|I|,t_a\}}\binom{|I|}{l}\binom{t_a}{l}$ to $(t_p+1)\binom{t_a+|I|}{t_a}$. As the cardinality of the intersection set I can range from 1 to $r-1$, we have the number of transmissions as $\sum_{|I|=1}^{r-1}\frac{\text{Number of mini-subfiles having } |I|}{(t_p+1)\binom{t_a+|I|}{t_a}}$ and the rate then becomes $\frac{\text{Number of transmissions}}{\binom{\Lambda}{t_a}\binom{\Lambda-t_a}{r}{t_p}}$. Therefore, we have

$$R=\sum_{|I|=1}^{r-1}\frac{\binom{\Lambda}{t_a}(t_p+1)\binom{\Lambda-t_a}{|I|}\sum_{j=0}^{r-|I|-1}(-1)^j\binom{\Lambda-t_a-|I|}{j}\binom{\binom{\Lambda-t_a-|I|-j}{r-|I|-j}}{t_p+1}}{\binom{\Lambda}{t_a}\binom{\binom{\Lambda-t_a}{r}{t_p}}(t_p+1)\binom{t_a+|I|}{t_a}},$$

which simplifies to

$$R=\sum_{|I|=1}^{r-1}\frac{\binom{\Lambda-t_a}{|I|}\sum_{j=0}^{r-|I|-1}(-1)^j\binom{\Lambda-t_a-|I|}{j}\binom{\binom{\Lambda-t_a-|I|-j}{r-|I|-j}}{t_p+1}}{\binom{\binom{\Lambda-t_a}{r}{t_p}}\binom{t_a+|I|}{t_a}}. \tag{4}$$

4.2 *Uniform-intersection subclass*

In this subsection, we prove Claim 2, which provides sufficient conditions under which a CMAP coded caching system belongs to the uniform-intersection subclass. We then characterize the corresponding achievable rate as stated in Corollary 1. Consider the following proof of Claim 2.

Proof. We begin the proof by defining N_k and C_k. Let $N_k=\binom{\Lambda}{t_a}\binom{\Lambda-t_a}{k}(t_p+1)\binom{\binom{\Lambda-t_a-k}{r-k}}{t_p+1}$ denote the number of demanded mini-subfiles having an intersection set of cardinality atleast k. Further, let

$$C_k=\binom{\Lambda}{t_a}(t_p+1)\binom{\Lambda-t_a}{k}\sum_{j=0}^{r-k-1}(-1)^j\binom{\Lambda-t_a-k}{j}\binom{\binom{\Lambda-t_a-k-j}{r-k-j}}{t_p+1}$$

denote the number of demanded mini-subfiles having an intersection set of cardinality exactly k. Observe that C_k can be written as:

$$C_k=\sum_{j=0}^{r-k-1}(-1)^j\binom{\Lambda-t_a-k}{j}N_{k+j}.$$

Since in the uniform-intersection subclass, all mini-subfiles have an intersection set of cardinality $|I|$, it follows that $N_k=0, k>|I|$. Therefore, we have

$$C_{|I|}=N_{|I|}+\sum_{j=1}^{r-k-1}(-1)^j\binom{\Lambda-t_a-k}{j}N_{|I|+j}$$

$$= \binom{\Lambda}{t_a}(t_p+1)\binom{\Lambda-t_a}{|I|}\binom{\binom{\Lambda-t_a-|I|}{r-|I|}}{t_p+1}. \tag{5}$$

Since $N_{|I|} \neq 0$, it follows that $t_p+1 \leq \binom{\Lambda-t_a-|I|}{r-|I|}$. However, as $N_k = 0, k \in [|I|+1, r-1]$, we have $t_p+1 > \binom{\Lambda-t_a-|I|-1}{r-|I|-1}$, yielding the second condition. Further, as every mini-subfile has an intersection set of cardinality $|I|$, the total number of demanded mini-subfiles equals $C_{|I|}$. Thus, we have

$$C_{|I|} = \binom{\Lambda}{r}\binom{\Lambda-r}{t_a}\binom{\binom{\Lambda-t_a}{r}-1}{t_p}. \tag{6}$$

Equating equations (5) and (6), we have $\binom{\Lambda}{t_a}\binom{\Lambda-t_a}{|I|}(t_p+1)\binom{\binom{\Lambda-t_a-|I|}{r-|I|}}{t_p+1} = \binom{\Lambda}{r}\binom{\Lambda-r}{t_a}\binom{\binom{\Lambda-t_a}{r}-1}{t_p}$. Using the property that $\binom{n}{k}\binom{n-k}{l} = \binom{n}{l}\binom{n-l}{k}$, we get $\binom{\Lambda-t_a}{|I|}(t_p+1)\binom{\binom{\Lambda-t_a-|I|}{r-|I|}}{t_p+1} = \binom{\Lambda-t_a}{r}\binom{\binom{\Lambda-t_a}{r}-1}{t_p}$. We obtain $\binom{\Lambda-t_a}{|I|}\binom{\Lambda-t_a-|I|}{r-|I|}\binom{\binom{\Lambda-t_a-|I|}{r-|I|}-1}{t_p} = \binom{\Lambda-t_a}{r}\binom{\binom{\Lambda-t_a}{r}-1}{t_p}$ using the property that $\binom{n+1}{k+1} = \frac{n+1}{k+1}\binom{n}{k}$. Finally, we simplify the condition to

$$\binom{r}{|I|}\binom{\binom{\Lambda-t_a-|I|}{r-|I|}-1}{t_p} = \binom{\binom{\Lambda-t_a}{r}-1}{t_p},$$

using the property that $\binom{\Lambda-t_a}{i}\binom{\Lambda-t_a-i}{r-i} = \binom{r}{i}\binom{\Lambda-t_a}{r}$. □

We now characterize the rate performance of the uniform-intersection subclass as given below.

Proof. Given that the conditions in Claim 2 are satisfied, all the demanded mini-subfiles have an intersection set of cardinality $|I| = i$. Thus, the rate becomes

$$R = \frac{\text{Number of mini-subfiles having } |I| = i}{\binom{\Lambda}{t_a}\binom{\binom{\Lambda-t_a}{r}}{t_p}(t_p+1)\binom{t_a+i}{t_a}} = \frac{\text{Number of demanded mini-subfiles}}{\binom{\Lambda}{t_a}\binom{\binom{\Lambda-t_a}{r}}{t_p}(t_p+1)\binom{t_a+i}{t_a}}.$$

Using this, we have,

$$R = \frac{\binom{\Lambda}{r}\binom{\Lambda-r}{t_a}\binom{\binom{\Lambda-t_a}{r}-1}{t_p}}{\binom{\Lambda}{t_a}\binom{\binom{\Lambda-t_a}{r}}{t_p}(t_p+1)\binom{t_a+i}{t_a}} = \frac{\binom{\Lambda-t_a}{r}\binom{\binom{\Lambda-t_a}{r}-1}{t_p}}{\binom{\binom{\Lambda-t_a}{r}}{t_p}(t_p+1)\binom{t_a+i}{t_a}}$$

$$= \frac{\binom{\Lambda-t_a}{r}}{(t_p+1)\binom{t_a+i}{t_a}}\left(1 - \frac{t_p}{\binom{\Lambda-t_a}{r}}\right) = \frac{\binom{\Lambda-t_a}{r} - t_p}{(t_p+1)\binom{t_a+i}{t_a}}, \tag{7}$$

□

Remark 4 In the uniform-intersection subclass, the cardinality of the intersection set is constant across all mini-subfiles. Consequently, the proposed scheme is regular in the sense that each coded transmission serves the same number of users. The resulting rate is given in Corollary 1, with coding gain $g = (t_p+1)\binom{t_a+i}{t_a}$, where the coding gain denotes the number of users served per transmission, and $i \in [1, r-1]$ is the intersection cardinality.

As t_p increases, the intersection size i tends to decrease. Although the factor (t_p+1) grows linearly with t_p, the term $\binom{t_a+i}{t_a}$ decreases rapidly as i decreases. Hence, the overall coding gain may deteriorate for larger values of t_p.

4.3 *Alpha bound*

In this subsection, we formulate the delivery phase of the scheme described in Subsection 4.1 as an ICP, following the approach in [21]. Specifically, we identify a set of mini-subfiles requested by users such that no user demanding a mini-subfile in the set possesses any other mini-subfile in the set as side information; this set forms a generalized independence set. The cardinality of the largest such set, denoted by $\alpha(\mathscr{I})$, provides a lower bound on the number of transmissions required to satisfy all user demands. Dividing this bound by the subpacketization F yields a lower bound on the optimal worst-case rate, as stated in Theorem 4.

We define U_i as the i^{th} user, $i \in \left[\binom{\Lambda}{r}\right]$, and U_l^S as the l^{th} user, demanding the subfile S, $l \in \left[\binom{\Lambda-t_a}{r}\right]$, under lexicographical ordering of the user-index sets.

We construct the set $B(\mathbf{d}) = B_1(\mathbf{d}) \cup B_2(\mathbf{d})$ whose elements are messages of the ICP $\mathscr{I}$ such that the set of indices of the messages in $B(\mathbf{d})$ forms a generalized independent set, where

$$B_1(\mathbf{d}) = \bigcup_{i\in[1,\Lambda-r-t_a+1]} \bigcup_{L\subset\left[i+1,\binom{\Lambda-ta}{r}\right],|L|=t_p} \{W_{d_{U_i},S,U_{l,l\in L}^S}, S\subset[r+i,\Lambda],|S|=t_a\},$$

and

$$B_2(\mathbf{d}) = \bigcup_{m\in\left[\Lambda-r-ta+2,\binom{\Lambda-ta}{r}\right]} \bigcup_{L\subset\left[m+1,\binom{\Lambda-ta}{r}\right],|L|=t_p} \{W_{d_{U_m^{S'}},S',U_{l,l\in L}^{S'}}\},$$

where $S' = [\Lambda+1-t_a,\Lambda]$.

Claim 3 The indices of the set $B(\mathbf{d})$ form a generalized independence set.

Proof. Let the set of indices of $B(\mathbf{d})$ be $H(\mathbf{d})$. Observe that each message in $B(\mathbf{d})$ is wanted by one receiver. Hence, all subsets of $H(\mathbf{d})$ of cardinality one are present in $\mathscr{J}(\mathscr{I})$. Next, consider a set

$$C = \{W_{d_{U_{i_1}},S_{j_1},U_{l,l\in L_{k_1}}^{S_{j_1}}},\cdots,W_{d_{U_{i_c}},S_{j_c},U_{l,l\in L_{k_c}}^{S_{j_c}}}\} \subseteq B(\mathbf{d}),$$

where $i_1 \le i_2 \le \cdots \le i_c$.

Consider the message $W_{d_{U_{i_1}},S_{j_1},U_{l,l\in L_{k_1}}^{S_{j_1}}}$. Since the receiver demanding this message does not have any other message in C as side information by construction of $B(\mathbf{d})$, it follows that the indices of messages in C belong to $\mathscr{J}(I)$ and any subset of $H(\mathbf{d})$ will lie in $\mathscr{J}(\mathscr{I})$. Thus, $H(\mathbf{d})$ forms a generalized independence set. □

Since $H(\mathbf{d})$ forms a generalized independence set, it follows that $\alpha(\mathscr{I}) \ge |H(\mathbf{d})| \implies \alpha(\mathscr{I}) \ge |B(\mathbf{d})|$ as $|H(\mathbf{d})| = |B(\mathbf{d})|$. Since the elements of $B_1(\mathbf{d})$ and $B_2(\mathbf{d})$ are disjoint, we have $|B(\mathbf{d})| = |B_1(\mathbf{d})| + |B_2(\mathbf{d})|$. We will now calculate $|B_1(\mathbf{d})|$.

Observe that a user U_i in the set $B_1(\mathbf{d})$ demands $\binom{\Lambda-r-i+1}{t_a}$ subfiles. Further, for each such subfile, user U_i demands $\binom{\binom{\Lambda-t_a}{r}-i}{t_p}$ mini-subfiles. Therefore, it follows that $|B_1(\mathbf{d})| = \sum_{i=1}^{\Lambda-t_a-r+1}\binom{\Lambda-r-i+1}{t_a}\binom{\binom{\Lambda-t_a}{r}-i}{t_p}$. We now compute $|B_2(\mathbf{d})|$.

Each user $U_m^{S'}$ in the set $B_2(\mathbf{d})$ demands $\binom{\binom{\Lambda-t_a}{r}-m}{t_p}$ mini-subfiles of the subfile S'. Therefore, $|B_2(\mathbf{d})| = \sum_{m=\Lambda-r-t_a+2}^{\binom{\Lambda-t_a}{r}} \binom{\binom{\Lambda-t_a}{r}-m}{t_p}$.

Hence, we conclude that

$$|B(\mathbf{d})| = \sum_{i=1}^{\Lambda-t_a-r+1} \binom{\Lambda-r-i+1}{t_a} \binom{\binom{\Lambda-t_a}{r}-i}{t_p} + \sum_{m=\Lambda-r-t_a+2}^{\binom{\Lambda-t_a}{r}} \binom{\binom{\Lambda-t_a}{r}-m}{t_p}.$$

Since $\alpha(\mathscr{S}) \geq |B(\mathbf{d})|$, the lower bound on the number of transmissions required to be made by the server in order to satisfy the demands of the users is as follows

$$T \geq \sum_{i=1}^{\Lambda-t_a-r+1} \binom{\Lambda-r-i+1}{t_a} \binom{\binom{\Lambda-t_a}{r}-i}{t_p} + \sum_{m=\Lambda-r-t_a+2}^{\binom{\Lambda-t_a}{r}} \binom{\binom{\Lambda-t_a}{r}-m}{t_p}.$$

For the given Λ, r, t_a and t_p, the subpacketization is fixed at $F = \binom{\Lambda}{t_a}\binom{\binom{\Lambda-t_a}{r}}{t_p}$. Dividing the lower bound on the number of transmissions by the subpacketization $F = \binom{\Lambda}{t_a}\binom{\binom{\Lambda-t_a}{r}}{t_p}$ yields the lower bound on the optimal worst-case rate stated in Theorem 4,

$$R_{UC}^{*}(M_a, M_p) = \frac{T}{\binom{\Lambda}{t_a}\binom{\binom{\Lambda-t_a}{r}}{t_p}} \geq \frac{\sum_{i=1}^{\Lambda-t_a-r+1} \binom{\Lambda-r-i+1}{t_a} \binom{\binom{\Lambda-t_a}{r}-i}{t_p} + \sum_{m=\Lambda-r-t_a+2}^{\binom{\Lambda-t_a}{r}} \binom{\binom{\Lambda-t_a}{r}-m}{t_p}}{\binom{\Lambda}{t_a}\binom{\binom{\Lambda-t_a}{r}}{t_p}}.$$

5. Numerical comparisons

This section presents numerical comparisons between the proposed scheme described in Section 4, the bounds summarized in Section 2.2, and the index coding-based lower bound in Theorem 4.

Uniform-intersection subclass: Figure 2 considers a CMAP system with $\Lambda = 10$ access caches, each of size $M_a = 4.5$, a library of $N = 45$ files, and access degree $r = 8$, resulting in $K = 45$ users. The private cache size M_p varies from 1 to 7, satisfying the sufficient conditions in Claim 2. In this regime, the cardinality of the intersection set is constant across all mini-subfiles, and the scheme is regular. As M_p increases, the delivery rate of the proposed scheme decreases and remains close to the index coding-based lower bound over the entire parameter range. Furthermore, when the total memory accessible to each user is matched, i.e., the CMACC scheme [8] operates with cache size $M_a + \frac{M_p}{r}$, the proposed scheme achieves lower rates, demonstrating the benefit of exploiting private-cache side information in the CMAP model.

Intersection class: Figure 3 considers a $(\Lambda = 10, r = 7, M_a = 12, N = 120)$ CMAP system with $K = \binom{10}{7} = 120$ users. The private cache size M_p ranges from 1 to 3, satisfying the condition in Claim 1 but not the uniform-intersection subclass conditions. In this non-regular regime, although the transmission structure varies across mini-subfiles, the proposed scheme continues to achieve rates close to the lower bound and outperforms the CMACC scheme [8] with equivalent accessible memory.

Overall, the numerical results demonstrate that the proposed scheme maintains near-optimal performance across both regular (uniform-intersection subclass) and non-regular (general intersection class) settings, highlighting the effectiveness of structured private-cache placement in combinatorial multi-access coded caching systems.

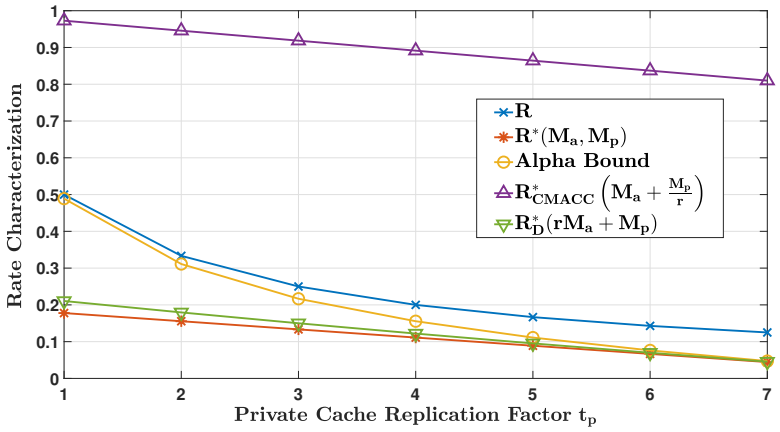


Figure 2. Rate vs t_p for a $(\Lambda = 10, r = 8, M_a = 4.5, M_p, N = 45)$-CMAP coded caching system

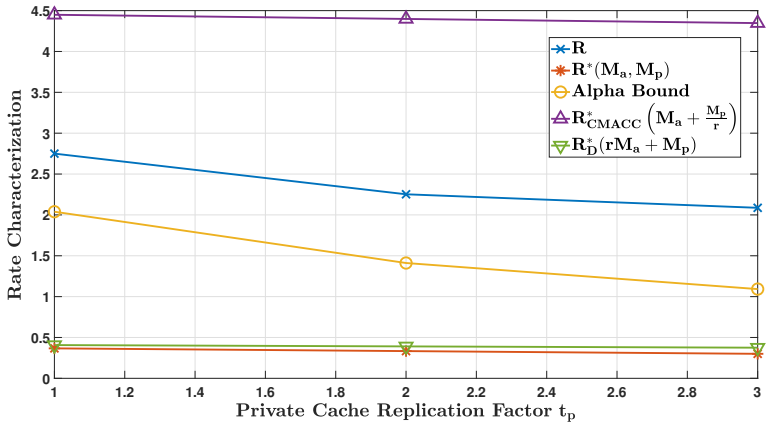


Figure 3. Rate vs t_p for a $(\Lambda = 10, r = 7, M_a = 12, M_p, N = 120)$-CMAP coded caching system

6. Conclusions

We introduced a constrained regime of the CMAP coded caching system, termed the intersection class, and derived a necessary and sufficient condition characterizing the system parameters under which this regime arises. For systems in this class, we proposed a centralized coded caching scheme and established its achievable rate-memory trade-off.

For the uniform-intersection subclass, we showed that the scheme admits a regular transmission structure, leading to a simplified rate expression and an explicit characterization of the coding gain. We further derived an index coding-based lower bound under the proposed uncoded placement, providing a benchmark for performance evaluation.

Numerical results demonstrate that the proposed scheme operates close to this lower bound across both uniform and general intersection regimes, highlighting the effectiveness of structured private-cache placement in combinatorial multi-access coded caching systems.

Acknowledgment

This work was supported partly by the Science and Engineering Research Board (SERB) of the Department of Science and Technology (DST), Government of India, through J.C. Bose National Fellowship to Prof. B. Sundar Rajan. In the

preparation of this work, the authors used ChatGPT for language editing and improving the clarity of presentation. After using this tool, the authors reviewed and edited the content as needed and take full responsibility for the content of the publication. All technical analysis, results, and conclusions were developed independently by the authors without the use of AI tools.

Conflicts of interest

The authors declare no competing financial interest.

References

[1] M. A. Maddah-Ali and U. Niesen, "Fundamental limits of caching," *IEEE Transactions on Information Theory*, vol. 60, no. 5, pp. 2856-2867, 2014. https://doi.org/10.1109/TIT.2014.2306938.

[2] K. Wan, D. Tuninetti, and P. Piantanida, "On the optimality of uncoded cache placement," In Proc. 2016 IEEE Information Theory Workshop, Cambridge, UK, Sep. 11-14, 2016, pp. 161-165. https://doi.org/10.1109/ITW.2016.7606816.

[3] Q. Yu, M. A. Maddah-Ali, and A. S. Avestimehr, "The exact rate-memory tradeoff for caching with uncoded prefetching," *IEEE Transactions on Information Theory*, vol. 64, no. 2, pp. 1281-1296, 2018. https://doi.org/10.1109/TIT.2017.2785237.

[4] M. Dutta and A. Thomas, "Decentralized coded caching for shared caches," *IEEE Communications Letters*, vol. 25, no. 5, pp. 1458-1462, 2021. https://doi.org/10.1109/LCOMM.2021.3052237.

[5] S. S. Meel and B. S. Rajan, "Secretive coded caching with shared caches," *IEEE Communications Letters*, vol. 25, no. 9, pp. 2849-2853, 2021. https://doi.org/10.1109/LCOMM.2021.3094405.

[6] 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. https://doi.org/10.1109/LCOMM.2016.2631552.

[7] F. Brunero and P. Elia, "Fundamental limits of combinatorial multi-access caching," *IEEE Transactions on Information Theory*, vol. 69, no. 2, pp. 1037-1056, 2023. https://doi.org/10.1109/TIT.2022.3193723.

[8] P. N. Muralidhar, D. Katyal, and B. S. Rajan, "Maddah-Ali-Niesen scheme for multi-access coded caching," In Proc. IEEE Information Theory Workshop, Kanazawa, Japan, Oct. 2021, pp. 1-6. https://doi.org/10.1109/ITW48936.2021.9611394.

[9] 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. https://doi.org/10.1109/TIT.2016.2557804.

[10] Y. Ma and D. Tuninetti, "On coded caching systems with offline users," In Proc. IEEE International Symposium on Information Theory, Espoo, Finland, Jun. 29-Jul. 1, 2022, pp. 1133-1138. https://doi.org/10.1109/ISIT50566.2022.9834866.

[11] X. Xie, K. Huang, J. Zhang, S. Gu, and Q. Zhang, "Coded caching in satellite networks," *IEEE Transactions on Communications*, vol. 73, no. 5, pp. 3227-3239, 2025. https://doi.org/10.1109/TCOMM.2024.3485596.

[12] M. Zhang, K. Wan, M. Cheng, and G. Caire, "Coded caching for two-dimensional multi-access networks," In Proc. 2022 IEEE International Symposium on Information Theory, Espoo, Finland, Jun. 26-Jul. 1, 2022, pp. 1707-1712. https://doi.org/10.1109/ISIT50566.2022.9834838.

[13] Y. Deng and M. Dong, "Fundamental structure of optimal cache placement for coded caching with nonuniform demands," *IEEE Transactions on Information Theory*, vol. 68, no. 10, pp. 6528-6547, 2022. https://doi.org/10.1109/TIT.2022.3179266.

[14] C. Gurjarpadhye, J. Ravi, S. Kamath, B. K. Dey, and N. Karamchandani, "Fundamental limits of demand-private coded caching," *IEEE Transactions on Information Theory*, vol. 68, no. 6, pp. 4106-4134, 2022. https://doi.org/10.1109/TIT.2022.3150336.

[15] K. Wan, M. Cheng, M. Kobayashi, and G. Caire, "On the optimal memory-load tradeoff of coded caching for location-based content," *IEEE Transactions on Communications*, vol. 70, no. 5, pp. 3047-3062, 2022. https://doi.org/10.1109/TCOMM.2022.3158326.

[16] C. Zhang, S. Wang, V. Aggarwal, and B. Peleato, "Coded caching with heterogeneous user profiles," *IEEE Transactions on Information Theory*, vol. 70, no. 3, pp. 1836-1847, 2024. https://doi.org/10.1109/TIT.2022.3186210.

[17] D. P. Singh, A. A. Mahesh, and B. S. Rajan, "Combinatorial multi-access coded caching with private caches," *arXiv*. https://arxiv.org/abs/2407.00677.

[18] Y. Birk and T. Kol, "Coding on demand by an informed source (ISCOD) for efficient broadcast of different supplemental data to caching clients," *IEEE Transactions on Information Theory*, vol. 52, no. 6, pp. 2825-2830, 2006. https://doi.org/10.1109/TIT.2006.874540.

[19] Z. Bar-Yossef, Y. Birk, T. S. Jayram, and T. Kol, "Index coding with side information," *IEEE Transactions on Information Theory*, vol. 57, no. 3, pp. 1479-1494, 2011. https://doi.org/10.1109/TIT.2010.2103753.

[20] S. H. Dau, V. Skachek, and Y. M. Chee, "Error correction for index coding with side information," *IEEE Transactions on Information Theory*, vol. 59, no. 3, pp. 1517-1531, 2013. https://doi.org/10.1109/TIT.2012.2227674.

[21] N. S. Karat, A. Thomas, and B. S. Rajan, "Error correction in coded caching with symmetric batch prefetching," *IEEE Transactions on Communications*, vol. 67, no. 8, pp. 5264-5274, 2019. https://doi.org/10.1109/TCOMM.2019.2916556.