An Index Coding Approach to Caching with Uncoded Cache Placement

Kai Wan, *Member, IEEE,* Daniela Tuninetti, *Senior Member, IEEE,* and Pablo Piantanida, *Senior Member, IEEE*

***Abstract*—Caching is an efficient way to reduce network traffic congestion during peak hours, by storing some content at the user's local cache memory, even without knowledge of user's later demands. Maddah-Ali and Niesen proposed a two-phase (placement phase and delivery phase) coded caching strategy for broadcast channels with cache-aided users. This paper investigates the same model under the constraint that content is placed uncoded within the caches, that is, when bits of the files are simply copied within the caches. When the cache contents are uncoded and the users' demands are revealed, the caching problem can be connected to an index coding problem. This paper focuses on deriving fundamental performance limits for the caching problem by using tools for the index coding problem that were either known or are newly developed in this work.**

First, a converse bound for the caching problem under the constraint of uncoded cache placement is proposed based on the "acyclic index coding converse bound." This converse bound is proved to be achievable by the Maddah-Ali and Niesen's scheme when the number of files is not less than the number of users, and by a newly derived index coding achievable scheme otherwise. The proposed index coding achievable scheme is based on distributed source coding and strictly improves on the widely used "composite (index) coding" achievable bound and its improvements, and is of independent interest.

An important consequence of the findings of this paper is that advancements on the coded caching problem posed by Maddah-Ali and Niesen are thus only possible by considering strategies with coded placement phase. A recent work by Yu *et al* has however shown that coded cache placement can at most half the network load compared to the results presented in this paper.

***Index Terms*—Coded caching; uncoded cache placement; index coding; distributed source coding.**

The results of this paper were presented in parts at the following conferences: the IEEE 2016 Information Theory Workshop, Cambridge, UK, the IEEE 2016 International Symposium on Information Theory, Barcelona, Spain, and the IEEE 2017 Information Theory and Applications Workshop, San Diego, CA USA. The results were also made available online as of Nov. 2015 as follows: arXiv:1511.02256, arXiv:1601.06383, and arXiv:1702.07265.

K. Wan was with Laboratoire de Signaux et Systèmes (L2S, UMR8506), CentraleSupélec-CNRS-Université Paris-Sud, 91192 Gif-sur-Yvette, France. He is now with Technische Universität Berlin, Berlin, Germany (email: kai.wan@tu-berlin.de).

D. Tuninetti is with the Electrical and Computer Engineering Department, University of Illinois at Chicago, Chicago, IL 60607 USA (e-mail: danielat@uic.edu).

P. Piantanida is with CentraleSupélec–French National Center for Scientific Research (CNRS)–Université Paris-Sud, 3 Rue Joliot-Curie, F-91192 Gif-sur-Yvette, France, and with Montreal Institute for Learning Algorithms (MILA) at Université de Montréal, 2920 Chemin de la Tour, Montréal, QC H3T 1N8, Canada (e-mail: pablo.piantanida@centralesupelec.fr).

The work of K. Wan was supported by Labex DigiCosme (project ANR11LABEX0045DIGICOSME) operated by ANR as part of the program "Investissement d'Avenir" Idex ParisSaclay (ANR11IDEX000302). The work of D. Tuninetti was supported in part by the National Science Foundation under award number 1527059. The work of P. Piantanida was supported by the European Commission's Marie Sklodowska-Curie Actions (MSCA), through the Marie Sklodowska-Curie IF (H2020-MSCAIF-2017-EF-797805-STRUDEL).

I. Introduction

NETWORKS have "rush hours," with peak traffic hours where traffic is high and the network performance suffers, and off-peak times, where traffic is low. Caching is an effective method to smooth out network traffic during peak times. In cache-aided networks, some content is locally stored into the users' local cache memory during off-peak hours in the hope that the pre-stored content will be required during peak hours. When this happens, content is retrieved locally thereby reducing the communication load, or number of transmissions, from the server to the users.

In this paper, we study the fundamental performance limits of cache-aided broadcast systems (also known as single bottleneck shared-link model) by following the model originally proposed by Maddah-Ali and Niesen (MAN) in their seminar works [1], [2]. We focus on the practically relevant case when content is stored uncoded in the local caches, in which case the caching problem can be related to the Index Coding (IC) problem. Although the connection between caching and IC is well known [1], [3], to the best of our knowledge, IC results have not been used to characterize the performance of the caching problem in the literature prior to our first work [4], which was publicly available online since November 2015. *This paper's main contribution is to leverage both known and hereby newly derived results for the IC problem to determine the fundamental limits of cache-aided systems with uncoded cache placement.* Since the publication of our work, a significant body of work on cache-aided systems has focused on characterizing the ultimate performance limits of caching schemes under the constraint of uncoded cache placement as we did in [4], [5]. A non-exhaustive list of such works includes Device-to-Device systems [6], coded caching systems with heterogeneous cache sizes [7], topological coded caching systems [8], coded caching systems with shared caches [9], coded data shuffling [10], etc.

A. Past Work

MAN's work: In [1], Maddah-Ali and Niesen proposed a coded caching scheme that utilizes an uncoded combinatorial cache construction in the placement phase and a binary linear network code in the delivery phase, where content in the caches is stored in a coordinated manner. The key observation is that well designed packets in the delivery phase are able to simultaneously satisfy many users at once, thus providing a "global caching gain" that scales with the total cache size in the network, in addition to the well known "local caching gain" that only depends on the amount of local cache at each user. In [1], Maddah-Ali and Niesen analytically showed that the load of their proposed scheme is to within a factor of 12 of a cut-set-type converse bound, but it was noted in [11] that it numerically appeared to be optimal to within a factor 4.7. The scheme in [1] has been improved in many ways; examples of schemes with uncoded cache placement are [4], [5], [12], [13], while with coded cache placement are [14]–[18].

Converse bounds for any cache placement: In [19]–[23], converse bounds tighter than cut-set bound provided in [1] and valid for any type of cache placement were proposed. An improved converse bound compared to the cut-set bound was given in [20, Theorem 1], which was used to show that the effectiveness of caching becomes small when the number of files becomes comparable to the square of the number of users. An algorithm that generalizes [20] was proposed in [19] to generate lower bounds on $\alpha R + \beta M$, for positive integers (α, β) and where R is the load and M the cache size, and used to show that the achievable load in [1] is optimal to within a factor 4. Another converse bound was obtained in [21] by leveraging [24, Theorem 17.6.1] as a 'symmetrization argument' over demands and used to show that the achievable load in [1] is optimal to within a factor 8; the converse bound applies to the case where users can request multiple files from the server as well. Inspired by converse results for caching systems over general degraded broadcast channels [25], the authors in [23] proposed achievable and converse bounds for the worst-case and the average loads that are to within a multiplicative gap of 2.315 and 2.507, respectively. An approach based on solving a linear program derived from the sub-modularity of entropy and simplified by leveraging certain inherent symmetries in the caching problem was put forth in [22] as a means to computationally generate converse bounds; the approach allowed to solve the case of $K = 2$ users and any number of files N, and gives at present the tightest bounds for problems with small K and N (beyond which the computational approach becomes practically unfeasible).

Converse bounds for uncoded cache placement: A different line of work that was initiated with our work in [4], where we asked the question of what would be the ultimate performance limit of cache-aided systems if one restricts the placement phase to be uncoded. In [4], we studied such a setting from the lens of IC. The IC had been connected to coded caching earlier in [1, p. 2865, Section VIII.A]: " [...] Now, for fixed *uncoded* content placement and for fixed demands, the delivery phase of the caching problem induces a so-called IC problem. However, it is important to realize that the caching problem actually consists of exponentially many parallel such IC problems, one for each of the N^K possible user demands. Furthermore, the IC problem itself is computationally hard to solve even only approximately. The main contribution of this paper is to design the content placement such that each of the exponentially many parallel IC problems has simultaneously an efficient and analytical solution." In [4], we analyzed the performance of the caching problem for fixed uncoded content placement and for fixed demands as an IC; by leveraging a known IC converse bound, and by carefully picking certain user demands, we explicitly characterized the worst case load as a function of certain parameters of the placement phase, which we (Fourier Motzkin) eliminated to find a closed form expression for the optimal load. This paper is the long, journal version, of our series of conference works that stared with [4].

The exact memory-load tradeoff for cache-aided systems under the constraint of uncoded cache placement was characterized in [13]. The converse bound in [13] is derived by a genie-based idea (instead of directly leveraging the acyclic IC converse bound as we did in [4], [5]), which is equivalent to our approach here – see the first remark Section III-D. The genie-based idea was also extended to the case of average load (as opposed to worst case load) with uniform independent and identically distributed demands. Our IC-based approach also extends to the same average lead setting – see the second remark in Section III-D.

Optimality for uncoded cache placement: By enhancing the cut-set-type converse bound by an additional non-negative term, the achievable load in [1] and its enhanced version in [13] were proved to be optimal to within a factor 2 [26]. This is, to the best of our knowledge, the sharpest known multiplicative gap, and implies that coded cache placement can at most half the network load compared to the results presented in [4] (and in this paper) and in [13].

Achievability and extensions: Much work has gone into improving the MAN caching scheme, especially in the small cache size regime where coded cache placement can outperform uncoded placement (as already noted in [1]). We shall not deal further into this line of work as it is not relevant to our work here.

The MAN caching scheme has been generalized to account for caching with nonuniform demands, multi-demands, shared caches, distinct file sizes or distinct cache sizes, online caching placement, hierarchical coded caching or other network topologies, device-to-device applications, secrete caching schemes, cache-aided systems with finite file size, etc. The results for these important practical scenarios are not discuss here as they are not directly relevant for our work. We note that the connection between caching and IC can be also leveraged in these systems, as we did for combination networks [27].

B. Main Contributions

Our main contributions, and how they compare to existing works, are summarized in Table I. In a nut shell, we focus on cache-aided systems with uncoded placement and study their performance by drawing connections to the IC problem. More precisely,

1) *Converse for cache-aided systems with uncoded cache placement.* In Section III, by exploiting [28, Corollary 1] for the IC problem, we derive a converse bound for the load in centralized cache-aided systems with uncoded cache placement. We show that it matches the load in [1] when there are more files than users.
2) *Novel IC achievable bound and its application to cache-aided systems.* In Section IV, we propose an IC achievable bound based on Han's coding scheme [29], Slepian-Wolf coding [30] and non-unique decoding [31]. This achievable scheme is shown to strictly outperform the composite (index) coding scheme, and is, to the best our

Table I
SUMMARY OF THE PAPER CONTRIBUTIONS.

Problem	Proposed results	Compared to the literature
Index coding	Novel achievable scheme	Strictly outperform the state-of-the-art in [28]
Caching	Novel converse bound for uncoded placement	Originally proposed in [4], [5], and later in [13]
Caching	Our novel IC bound achieves our novel converse	A generalized version of the scheme in [13] from the viewpoint of index coding

knowledge, the best random coding achievable bound for the general IC problem to date. We then show that our novel IC scheme matches our converse bound for centralized cache-aided systems with uncoded cache placement when there are less files than users, and attains the same load as the scheme in [26].

C. Paper Outline

The rest of the report is organized as follows. The system models for centralized cache-aided systems and for IC, and their relationship, are introduced in Section II, as well as the results on those two problems needed in later sections. In Section III, we derive a converse bound under the constraint of uncoded cache placement. In Section IV, we introduce a novel IC achievable bound and use it to design a caching scheme that achieves the proposed converse bound for the caching problem. Section V concludes the paper. Some proofs may be found in Appendix.

Results for decentralized cache-aided systems[1] under the constraint of uncoded cache placement are not reported here because they follow from the same line of reasoning as those for centralized systems as detailed in [32] – see also the third remark in Section IV-C.

II. SYSTEM MODELS AND SOME KNOWN RESULTS

A. Notation

Calligraphic symbols denote sets; symbols in bold font denote vectors; $|\cdot|$ is used to represent the cardinality of a set or the length of a file in bits; we let $\mathcal{A}\setminus\mathcal{B} := \{x \in \mathcal{A} | x \notin \mathcal{B}\}$, $[a:b:c] := \{a, a+b, a+2b, ..., c\}$, $[a:c] = [a:1:c]$ and $[n] = [1:n]$; the bit-wise XOR operation between binary vectors is indicated by $\oplus$; for two integers x and y, we let $\binom{x}{y} = 0$ if $x < y$ or $x \le 0$.

B. The Centralized Caching Problem: Definition

The information-theoretic formulation of the centralized coded caching problem in Fig. 1, as originally formulated by Maddah-Ali and Niesen in [1], is as follows.

- The system comprises a server with N independent files, denoted by $(F_1, F_2, \dots, F_N)$, and K users connected to it through an error-free link. Each file has B independent and equally likely bits.
- In the placement phase, user $k \in [K]$ stores content from the N files in its cache of size MB bits without knowledge of later demands, where $M \in [0, N]$. We denote the content in the cache of user $k \in [K]$ by $Z_k = \phi_k(F_1, \dots, F_N)$, where

$$\phi_k : [0:1]^{NB} \to [0:1]^{\lfloor MB \rfloor}, \ \forall k \in [K]. \qquad (1)$$

  We also denote by $\mathbf{Z} := (Z_1, \dots, Z_K)$ the content of all the caches.
- In the delivery phase, each user demands one file and the demand vector $\mathbf{d} := (d_1, d_2, \dots, d_K)$, where $d_k \in [N]$ corresponds to the file demanded by user $k \in [K]$, is revealed to the server and all users. Given $(\mathbf{Z}, \mathbf{d})$, the server broadcasts the message $X_{\mathbf{d}} = \psi(F_1, \dots, F_N, \mathbf{d})$, where

$$\psi : [0:1]^{NB} \times [N]^K \to [0:1]^{\lceil RB \rceil}. \qquad (2)$$

- Each user $k \in [K]$ estimates the demanded file as $\widehat{F}_k = \mu_k(X_{\mathbf{d}}, Z_k)$, where

$$\mu_k : \ [0:1]^{\lceil RB \rceil} \times [0:1]^{\lfloor MB \rfloor} \to [0:1]^B, \ \forall k \in [K]. \qquad (3)$$

- The (worst-case over all possible demands) probability of error is

$$P_{\mathrm{e}}^{(B)} := \max_{\mathbf{d} \in [N]^K} \Pr\left[\bigcup_{k=1}^{K} \left\{ \widehat{F}_k \neq F_{d_k} \right\} \right]. \qquad (4)$$

- A pair (M, R) is said to be achievable if there exit placement functions as in (1), encoding function as in (2) and decoding functions as in (3) such that $\lim_{B\to\infty} P_{\mathrm{e}}^{(B)} = 0$, where $P_{\mathrm{e}}^{(B)}$ was defined in (4).
- The objective is to determine, for a fixed M, the (worst-case) load

$$R^\star := \inf\{R : (M, R) \text{ is achievable}\}. \qquad (5)$$

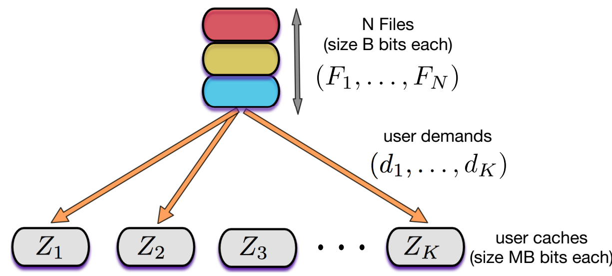

Figure 1. A centralized cache-aided system where a server with N files of size B bits is connected to K users equipped with a cache of size MB bits.

[1]Cache-aided systems are divided into two classes, *centralized* [1] and *decentralized* [2], depending on whether users can coordinate during the placement phase. In centralized cache-aided systems, the users in the two phases of the caching scheme are assumed to be the same; therefore, coordination among users is possible in the placement phase. In practice, for example due to users' mobility, a user may be connected to a server during its placement phase but to a different one during its delivery phase; in such scenarios, coordination among users during the placement phase is thus not possible.

In the following, we say that the placement phase is *uncoded* if the bits of the various files are simply copied within the caches. Formally,

Definition 1 (Uncoded cache placement)**.** *The placement phase is said to be uncoded if the cache contents in* (1) *are* $Z_k = (A_{1,k}, A_{2,k}, \dots, A_{N,k})$ *where* $A_{i,k} \subseteq F_i$ *for all files* $i \in [N]$ *and such that* $\sum_{i\in[N]} |A_{i,k}| \leq MB$*, for all users* $k \in [K]$*.*

The (worst-case) load under the constraint of uncoded cache placement is denoted as $R^\star_{\text{u}}$. Trivially $R^\star_{\text{u}} \geq R^\star$.

C. The Centralized Caching Problem: MAN Achievability

We start with the description of the MAN scheme in [1]. Let the cache size be $M = t\frac{N}{K}$, for some positive integer $t \in [0:K]$. In the placement phase, each file is partitioned into $\binom{K}{t}$ equal size sub-files of $B/\binom{K}{t}$ bits. The sub-files of F_i are denoted by $F_{i,\mathcal{W}}$ for $\mathcal{W} \subseteq [K]$ where $|\mathcal{W}| = t$. User $k \in [K]$ fills its cache as

$$Z_k = \Big(F_{i,\mathcal{W}} : k \in \mathcal{W},\ \mathcal{W} \subseteq [K],\ |\mathcal{W}| = t,\ i \in [N]\Big). \quad (6)$$

In the delivery phase, given the demand vector $\mathbf{d}$, the server transmits

$$X_{\mathbf{d}} = \Big(\oplus_{s\in\mathcal{S}} F_{d_s, \mathcal{S}\setminus\{s\}} : \mathcal{S} \subseteq [K], |\mathcal{S}| = t+1\Big), \quad (7)$$

which requires broadcasting $B\binom{K}{t+1}/\binom{K}{t}$ bits. Note that user $k \in \mathcal{S}$, for $\mathcal{S}$ as in (7), wants $F_{d_k,\mathcal{S}\setminus\{k\}}$ and has cached $F_{d_s,\mathcal{S}\setminus\{s\}}$ for all $s \in \mathcal{S} : s \neq k$, so it can recover $F_{d_k,\mathcal{S}\setminus\{k\}}$ from $X_{\mathbf{d}}$ in (7) and the cache content in (6). The load is thus given by

$$R_{\text{c,u,MAN}}[t] := \frac{\binom{K}{t+1}}{\binom{K}{t}} \geq R^\star, \quad (8)$$

where the subscript "c,u,MAN" in (8) stands for "centralized uncoded-placement Maddah-Ali and Niesen." For $M\frac{K}{N}$ not an integer, one takes the lower convex envelope of the set of points $(M, R) = \big(t\frac{N}{K}, R_{\text{c,u,MAN}}[t]\big)$ for $t \in [0:K]$.

The MAN scheme was improved in [26] as follows. Of the $\binom{K}{t+1}$ transmitted linear combinations in (7), $\binom{K-\min(K,N)}{t+1}$ can be obtained as linear combinations of the other transmissions. Therefore, by removing these redundant transmissions for the case $N < K$, the load becomes

$$R_{\text{c,u,YMA}}[t] := \frac{\binom{K}{t+1} - \binom{K-\min(K,N)}{t+1}}{\binom{K}{t}} \geq R^\star, \quad (9)$$

for $M = t\frac{N}{K}$ with $t \in [0:K]$, where the subscript "c,u,YMA" in (9) stands for "centralized uncoded-placement Yu Maddah-Ali Avestimehr." For $M\frac{K}{N}$ not an integer, one takes the lower convex envelope of set of points $(M, R) = \big(t\frac{N}{K}, R_{\text{c,u,YMA}}[t]\big)$ for $t \in [0:K]$. Notice that the load in (9) is strictly smaller than the one in (8) for $N < K$.

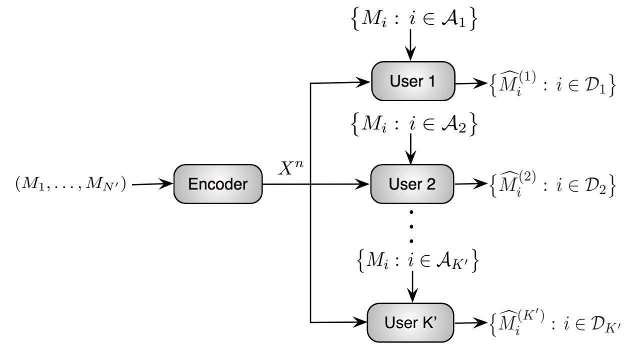

Figure 2. An IC problem with N' files and K' users.

D. The Index Coding Problem: Definition

The IC problem, shown in Fig. 2 and originally proposed in [33] in the context of broadcasting with message side information, is defined as follows.

- A sender wishes to communicate N' independent messages to K' users. The server is connected to the users through a noiseless channel with finite input alphabet $\mathcal{X}$.
- Each user $j \in [K']$ demands a set of messages indexed by $\mathcal{D}_j \subseteq [N']$ and knows a set of messages indexed by $\mathcal{A}_j \subseteq [N']$. In order to avoid trivial problems, it is assumed that $\mathcal{D}_j \neq \emptyset$, $\mathcal{A}_j \neq [N']$, and $\mathcal{D}_j \cap \mathcal{A}_j = \emptyset$.
- A $(|\mathcal{X}|^{nR_1}, \dots, |\mathcal{X}|^{nR_{N'}}, n, \epsilon_n)$-code for the IC problem is defined as follows. Each message M_i, $i \in [N']$, is uniformly distributed on $[|\mathcal{X}|^{nR_i}]$, where n is the block-length and $R_i \geq 0$ is the transmission rate in symbols per channel use. In order to satisfy the users' demands, the server broadcasts $X^n = \mathsf{enc}(M_1, \dots, M_{N'}) \in \mathcal{X}^n$ where enc is the encoding function. Each user $j \in [K']$ estimates the messages indexed by $\mathcal{D}_j$ by the decoding function $\mathsf{dec}_j\big(X^n, (M_i : i \in \mathcal{A}_j)\big)$. The probability of error is

$$\epsilon_n := \max_{j\in[K']} \Pr\big[\mathsf{dec}_j\big(X^n, (M_i : i \in \mathcal{A}_j)\big) \neq (M_i : i \in \mathcal{D}_j)\big]. \quad (10)$$

- A rate vector $(R_1, \dots, R_{N'})$ is said to be achievable if there exists a family of $(|\mathcal{X}|^{nR_1}, \dots, |\mathcal{X}|^{nR_{N'}}, n, \epsilon_n)$-codes for which $\lim_{n\to\infty} \epsilon_n = 0$, for ϵ_n in (10).
- The goal is to find the capacity region, defined as the largest possible set of achievable rate vectors.

Remark 1. *We used the definitions in [34, Chapter 1, Section 1.2], in which the definition of capacity region depends on the alphabet size* $|\mathcal{X}|$*. However, as proved in [34, Lemma 1.1], the choice of the alphabet size* $\mathcal{X}$ *is irrelevant to the actual capacity region itself. Intuitively, this is so because the rates are defines in symbols per channel use or, equivalently, the base of the logarithms is* $|\mathcal{X}|$*.*

E. The Index Coding Problem: Composite (Index) Coding Achievable Bound

The composite (index) coding achievable bound proposed in [28] is a two-stage scheme based on binning and non-unique decoding. In the first encoding stage, for each $\mathcal{J} \subseteq [N']$, the messages $(M_i : i \in \mathcal{J})$ are encoded into the *composite index* $W_{\mathcal{J}} \in [|\mathcal{X}|^{nS_{\mathcal{J}}}]$ at some rate $S_{\mathcal{J}} \geq 0$ based on random binning. By convention, $S_{\emptyset} = 0$. In the second encoding stage, the collection of all composite indices $(W_{\mathcal{J}} : \mathcal{J} \subseteq [N'])$ is mapped into a length-n sequence X^n which is received error-free by all users.

In the first decoding stage, every user recovers all composite indices. In the second decoding stage, user $j \in [K']$ chooses a set $\mathcal{K}_j$ such that $\mathcal{D}_j \subseteq \mathcal{K}_j \subseteq [N'] \backslash \mathcal{A}_j$ and uniquely decodes all messages $(M_i : i \in \mathcal{K}_j)$; the decoding of user $j \in [K']$ is based on the recovered $(W_{\mathcal{J}} : \mathcal{J} \subseteq \mathcal{K}_j \cup \mathcal{A}_j)$.

The achievable rate region by composite (index) coding is stated next, from [28, Proposition 6.11].

Theorem 1 (Composite IC Achievable Bound, generalized to allow for multicast messages)**.** *A non-negative rate tuple* $\mathbf{R} := (R_1, \ldots, R_{N'})$ *is achievable for the IC problem* $\big((\mathcal{A}_j, \mathcal{D}_j) : j \in [K']\big)$ *defined in Section II-D provided that*

$$\mathbf{R} \in \bigcap_{j \in [K']} \bigcup_{\mathcal{K}_j : \mathcal{D}_j \subseteq \mathcal{K}_j \subseteq [N'] \backslash \mathcal{A}_j} \mathscr{R}(\mathcal{K}_j | \mathcal{A}_j, \mathcal{D}_j), \tag{11a}$$

$$\mathscr{R}(\mathcal{K} | \mathcal{A}, \mathcal{D}) := \bigcap_{\mathcal{J} : \mathcal{J} \subseteq \mathcal{K}} \left\{ \sum_{i \in \mathcal{J}} R_i < v_{\mathcal{J}} \right\}, \tag{11b}$$

where in (11b) $v_{\mathcal{J}}$ *is defined as*

$$v_{\mathcal{J}} := \sum_{\mathcal{P} : \mathcal{P} \subseteq \mathcal{A} \cup \mathcal{K}, \mathcal{P} \cap \mathcal{J} \neq \emptyset} S_{\mathcal{P}}, \tag{11c}$$

and where in (11c) *the non-negative quantities* $(S_{\mathcal{J}} : \mathcal{J} \subseteq [N'])$ *must satisfy*

$$\sum_{\mathcal{J} : \mathcal{J} \in [N'], \mathcal{J} \nsubseteq \mathcal{A}_j} S_{\mathcal{J}} < 1, \quad \forall j \in [K']. \tag{11d}$$

Note that the constrain in (11d) is from the first decoding stage and the one in (11c) from the second decoding stage.

F. The Index Coding Problem: Acyclic Subgraph Converse Bound for Multiple Unicast Index Coding

If $K' = N'$ and $\mathcal{D}_j = \{j\}$ where $j \notin \mathcal{A}_j$ for each $j \in [N']$, the IC is known as the *multiple unicast IC*. The multiple unicast IC problem can be represented as a directed graph G, where each node in the graph represents one user and its demanded message, and where a directed edge connects node i to node j if user j knows the message desired by user i. By the submodularity of entropy, a converse bound was proposed in [35, Theorem 3.1] for the symmetric rate case and extended in [28, Theorem 1] to the case where messages can have different rates. Due to the high computational complexity, the converse bound in [28, Theorem 1] can only be evaluated for IC problems with limited number of messages. A looser (compared to [28, Theorem 1]) converse bound was proposed in [28, Corollary 1] and is stated next.

Theorem 2 (Acyclic Subgraph Converse Bound for Multiple Unicast IC [28])**.** *If* $(R_1, \ldots, R_{N'})$ *is achievable for the multiple unicast IC problem* $\big((\mathcal{A}_j, \mathcal{D}_j = \{j\}) : j \in [N']\big)$, *defined in Section II-D and represented by the directed graph G, then it must satisfy*

$$\sum_{j \in \mathcal{J}} R_j \leq 1, \tag{12}$$

for all $\mathcal{J} \subseteq [N']$ *where the sub-graph of G over the vertices in* $\mathcal{J}$ *does not contain a directed cycle.*

The proof of Theorem 2 is based on noticing that a user k_1 in the found acyclic subgraph can decode the message of the following user k_2 either because $\mathcal{A}_{k_2} \subseteq \mathcal{D}_{k_1} \cup \{k_1\}$ (user k_1 can mimic user k_2) or by giving user k_1 genie side information so that it can mimic user k_2.

G. Mapping the Caching Problem with Uncoded Cache Placement into an Index Coding Problem

As mentioned before, the caching problem with uncoded cache placement can be see as a family of IC problems. The difference between caching and IC is that the side information sets are fixed in IC, while they represent the cache contents that must be properly designed in caching; moreover, in IC the demands are also fixed, while in caching one must consider all possible demands. In caching, if the cache placement phase is uncoded, the delivery phase is an IC problem for appropriately defined message and demand sets. Hence, IC results can be leveraged for the caching problem, as we do in this paper.

Under the constraint of uncoded cache placement, when the cache contents and the demands are fixed, the delivery phase of the caching problem is equivalent to the following IC problem. Denote the set of distinct demanded files in the demand vector $\mathbf{d}$ by $\mathcal{N}(\mathbf{d})$, that is $\mathcal{N}(\mathbf{d}) := \cup_{k \in [K]} \{d_k\}$. For each $i \in \mathcal{N}(\mathbf{d})$ and for each $\mathcal{W} \subseteq [K]$, the sub-file $F_{i,\mathcal{W}}$ (containing the bits of file F_i only within the cache of the users indexed by $\mathcal{W}$) is an independent message in the IC problem with user set $[K]$. Hence, by using the notation introduced in Sections II-D and II-B, we have $K' = K$ and $N' \leq |\mathcal{N}(\mathbf{d})|(2^K - 1)$. For each user $k \in [K]$ in this general IC problem, the desired message set and the side information sets are given by

$$\mathcal{D}_k := \big\{F_{d_k, \mathcal{W}} : \mathcal{W} \subseteq [K], k \notin \mathcal{W}\big\}, \tag{13}$$

$$\mathcal{A}_k := \big\{F_{i, \mathcal{W}} : \mathcal{W} \subseteq [K], i \in \mathcal{N}(\mathbf{d}), k \in \mathcal{W}\big\}. \tag{14}$$

III. Converse Bound for Centralized Cache-aided Systems with Uncoded Cache Placement

In this section, we leverage the connection between caching and IC problems outlined in Section II-G to investigate the fundamental limits of centralized cache-aided systems with uncoded cache placement. We derive a converse bound (by using Theorem 2 in Section II-F) that matches the achievable load $R_{\text{c,u,YMA}}$ in (9).

A. Theorem Statement

The following converse bound on the load of centralized cache-aided systems under the constraint of uncoded cache placement was first presented in our conference papers [4], [5]. Recall that we denote the optimal load by $R^{\star}$, and the optimal load under the constraint of uncoded cache placement (see Definition 1) as $R^{\star}_{\text{c,u}}$.

Theorem 3. *In centralized cache-aided systems the load $R^{\star}_{c,u}$ satisfies*

$$R^{\star}_{c,u} \geq c_q + (c_q - c_{q-1})\left(\frac{KM}{N} - q\right), \tag{15}$$

$$c_q := \frac{\binom{K}{q+1} - \binom{K-\min(K,N)}{q+1}}{\binom{K}{q}}, \quad \forall q \in [0:K]. \tag{16}$$

Moreover, this converse bound is a piece-wise linear curve with corner points

$$(M, R) = \left(q\frac{N}{K}, c_q\right), \quad \forall q \in [0:K]. \tag{17}$$

Before we proceed to prove the general converse bound in Theorem 3, we give a specific example. This example introduces the main ideas in the proof.

B. An Example

The reasoning in this example applies to the general case $K \leq N$; the case $K > N$ will be dealt with in the general proof.

Assume that the server has $N = K = 3$ files, denoted as (F_1, F_2, F_3). The file length in number of bits is B. The cache size in number of bits is MB, for some $M \in [0, N] = [0, 3]$. After the uncoded cache placement phase is done, each file F_i can be thought as having been divided into $2^K = 2^3 = 8$ disjoint sub-files, denoted as

$$\big(F_{i,\mathcal{W}} : \mathcal{W} \subseteq [K],\ i \in [N]\big), \tag{18}$$

where $F_{i,\mathcal{W}}$ has been cached only by the users indexed by $\mathcal{W}$. For simplicity in the following we omit the braces when we indicate sets, i.e., $F_{1,12}$ represents $F_{1,\{1,2\}}$, which dos not create any confusion for this example.

For each demand vector $\mathbf{d} = (d_1, d_2, d_3) \in [N]^K = [3]^3$, we generate an IC problem with at most $|\mathcal{N}(\mathbf{d})| 2^{K-1} = 12$ independent messages. These messages are

$$\bigcup_{k\in[K], \mathcal{W}\subseteq[K] : k\notin\mathcal{W}} F_{d_k,\mathcal{W}}, \tag{19}$$

and represents the sub-files demanded by the users in $[K]$ but not available in their caches. The messages available as side information to user $k \in [K]$ for this IC are

$$\bigcup_{i\in\mathcal{N}(\mathbf{d}), \mathcal{W}\subseteq[K] : k\in\mathcal{W}} F_{i,\mathcal{W}}. \tag{20}$$

For this IC problem, we generate a directed graph as follows. Each vertex corresponds to a different sub-file. There is a directed arrow from $F_{d_{k_1},\mathcal{W}_1}$ to $F_{d_{k_2},\mathcal{W}_2}$ if and only if user k_2 caches $F_{d_{k_1},\mathcal{W}_1}$ (i.e., $k_2 \in \mathcal{W}_1$). For example, Fig. 3 shows the directed graph representing this IC problem for the demand vector $\mathbf{d} = (1, 2, 3)$.

Consider the demand vector $\mathbf{d} = (d_1, d_2, d_3)$, where $d_i \in [N] = [3],\ i \in [K] = [3]$. In order to apply Theorem 2, in the constructed directed graph we want to find sets of vertices $\mathcal{J}$ that do not form a directed cycle. No receiver has stored $F_{1,\emptyset}, F_{2,\emptyset}, F_{3,\emptyset}$, so there is no outgoing edge from $F_{1,\emptyset}, F_{2,\emptyset}, F_{3,\emptyset}$ to any other vertex in the graph. Therefore, $F_{1,\emptyset}, F_{2,\emptyset}, F_{3,\emptyset}$ are always in the such sets $\mathcal{J}$ when we evaluate (12).

We focus next on demand vectors $\mathbf{d}$ with distinct demands, that is, $|\mathcal{N}(\mathbf{d})| = \min(N, K) = K = 3$; the worst case demand may not be in such a set of demand vectors, but this is not a problem as we aim to derive a converse bound on the (worst-case) load at this point. For a demand vector $\mathbf{d}$ with distinct demands, consider now a permutation $\mathbf{u} = (u_1, u_2, u_3)$ of $[K] = [3]$. For each such $\mathbf{u}$, a set of nodes not containing a cycle is as follows: $F_{d_{u_1},\mathcal{W}_1}$ for all $\mathcal{W}_1 \subseteq [K]\backslash\{u_1\}$, and $F_{d_{u_2},\mathcal{W}_2}$ for all $\mathcal{W}_2 \subseteq [K]\backslash\{u_1, u_2\}$, and $F_{d_{u_3},\mathcal{W}_3}$ for all $\mathcal{W}_3 \subseteq [K]\backslash\{u_1, u_2, u_3\} = \emptyset$. For example, when $\mathbf{d} = (1, 2, 3)$ and $\mathbf{u} = (1, 3, 2)$, we have

$$d_{u_1} = d_1 = 1; \mathcal{W}_1 \subseteq [K]\backslash\{u_1\} = [3]\backslash\{1\} = \{2, 3\}, \tag{21}$$

$$d_{u_2} = d_3 = 3; \mathcal{W}_2 \subseteq [K]\backslash\{u_1, u_2\} = [3]\backslash\{1, 3\} = \{2\}, \tag{22}$$

$$d_{u_3} = d_2 = 2; \mathcal{W}_3 \subseteq [K]\backslash\{u_1, u_2, u_3\} = \emptyset, \tag{23}$$

and the corresponding set not containing a cycle is

$$(F_{1,\emptyset}, F_{1,2}, F_{1,3}, F_{1,23}, F_{3,\emptyset}, F_{3,2}, F_{2,\emptyset}). \tag{24}$$

More precisely, $F_{1,\emptyset}, F_{1,2}, F_{1,3}, F_{1,23}$ are demanded by user 1 and thus there is no cycle among them. User 1 does not know $F_{3,\emptyset}, F_{3,2}$ and thus there is no directed arrow from each of $F_{3,\emptyset}, F_{3,2}$ to each of $F_{1,\emptyset}, F_{1,2}, F_{1,3}, F_{1,23}$. So there is no cycle in $(F_{1,\emptyset}, F_{1,2}, F_{1,3}, F_{1,23}, F_{3,\emptyset}, F_{3,2})$. Finally, user 1 and user 3 do not know $F_{2,\emptyset}$, and thus there is no directed arrow from $F_{2,\emptyset}$ to each of $F_{1,\emptyset}, F_{1,2}, F_{1,3}, F_{1,23}, F_{3,\emptyset}, F_{3,2}$. So we can see the chosen set is acyclic (see figure 4).

From (12), we have that the acyclic set of nodes in (24) can be used to write the following bound (in which $2^{BR^{\star}_{\text{c,u}}}$ plays the role of $|\mathcal{X}|$)

$$BR^{\star}_{\text{c,u}} \geq |F_{1,\emptyset}| + |F_{1,2}| + |F_{1,3}| + |F_{1,23}| + |F_{3,\emptyset}| + |F_{3,2}| + |F_{2,\emptyset}|. \tag{25}$$

In general, when $K \leq N$ we can find a bound such as the one in (25) for all possible pairs $\mathbf{d} \in \text{Perm}(N, K)$ and $\mathbf{u} \in \text{Perm}(K, K)$, where $\text{Perm}(n, k)$ denotes the set of all $k-$permutations of n elements (there are $\frac{n!}{(n-k)!}$ elements in the set $\text{Perm}(n, k)$ for $n \geq k$). If we sum all the $|\text{Perm}(N, K)||\text{Perm}(K, K)| = \binom{N}{K}(K!)^2 = (3!)^2 = 36$ inequalities as in (25), we get

$$R^{\star}_{\text{c,u}} \geq \frac{1}{(3!)^2} \sum_{\mathbf{d}\in\text{Perm}(3,3)} \sum_{\mathbf{u}\in\text{Perm}(3,3)} \sum_{j\in[3]} \sum_{\mathcal{W}_j\subseteq[3]\backslash\{u_1,\ldots,u_j\}} \frac{|F_{d_{u_j},\mathcal{W}_j}|}{B} \tag{26a}$$

$$= \sum_{t\in[0:3]} x_t \frac{\binom{3}{t+1}}{\binom{3}{t}} \tag{26b}$$

$$= 3 \cdot x_0 + 1 \cdot x_1 + \frac{1}{3} \cdot x_2 + 0 \cdot x_3, \tag{26c}$$

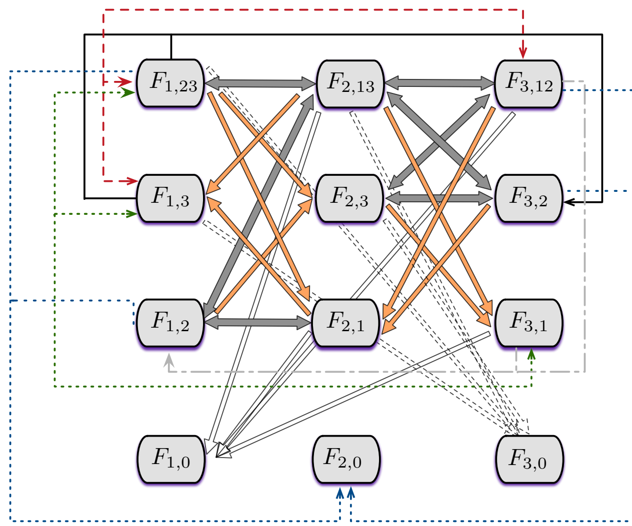

Figure 3. Directed graph for the equivalent IC scenario corresponding to the caching problem with $N = K = 3$, and with demand vector $\mathbf{d} = (1, 2, 3)$. Each sub-file demanded by each user is an independent node in the directed graph. A directed arrow from $F_{d_{k_1},\mathcal{W}_1}$ to $F_{d_{k_2},\mathcal{W}_2}$ appears if and only if user k_2 caches $F_{d_{k_1},\mathcal{W}_1}$ (i.e., $k_2 \in \mathcal{W}_1$). Different types and colours for arrows are intended to distinguish dense arrows in the figure.

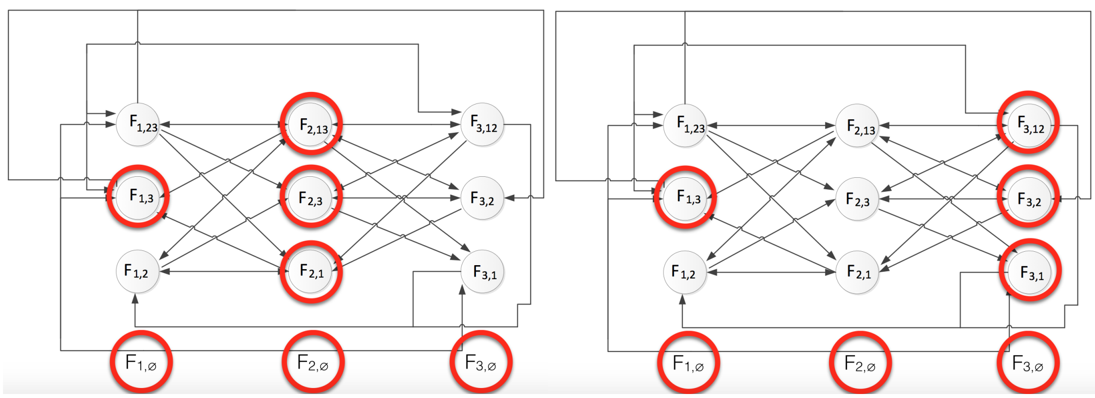

Figure 4. Nodes forming an acyclic subgraph are circled in red. Left: demand $(1, 2, 3)$ and permutation $(2, 1, 3)$. Right: demand $(1, 2, 3)$ and permutation $(3, 1, 2)$.

where x_t in (26b) is defined as

$$0 \le x_t := \sum_{j\in[N]} \sum_{\mathcal{W}\subseteq[K]:|\mathcal{W}|=t} \frac{|F_{j,\mathcal{W}}|}{NB}, \quad t \in [0:K], \tag{27}$$

and represnts the fraction of the total number of bits across all files that are known/cached exclusively by a subset of $t \in [0:K] = [0:3]$ users.

The general proof of (26b) can be found in (36b). At this point we can offer the following intuitive interpretation for the case $K \le N$, as it is the case in this example. The total number of sub-files cached by a subset of $t \in [0:K]$ users is $N\binom{K}{t}$, where the factor $\binom{K}{t}$ appears at the denominator of (26b) (here $K = 3$), and the factor N at the denominator of (27). The total number of sub-files cached by a subset of $t \in [0:K]$ users in (25) (by the symmetry of the problem, the other bounds for different pair $\mathbf{d}$ and $\mathbf{u}$ have the same structure as (25)) is

$$\sum_{i\in[K]} \sum_{\mathcal{W}_i\subseteq[K]\setminus\{u_1,\ldots,u_i\}} 1_{\{|\mathcal{W}_i|=t\}}$$
$$= \binom{K-1}{t} + \binom{K-2}{t} + \ldots + \binom{t}{t} = \binom{K}{t+1}, \tag{28}$$

where $1_{\{\mathcal{A}\}}$ in the indicator function that is equal to one if and only if the condition in $\mathcal{A}$ is true, and where we use the Pascal's triangle identity; the factor $\binom{K}{t+1}$ (here $K = 3$) appears at the numerator of (26b).

In addition to the bounds in (26) and (27), we also have that the total number of bits in the files is

$$\sum_{j\in[N]} \sum_{\mathcal{W}\subseteq[K]} |F_{j,\mathcal{W}}| = NB \Longleftrightarrow \sum_{t\in[K]} x_t = 1, \tag{29}$$

and that the total number of bits in the caches must satisfy

$$\sum_{j\in[N]} \sum_{\mathcal{W}\subseteq[K]:j\in\mathcal{W}} |F_{j,\mathcal{W}}| \le KMB \Longleftrightarrow \sum_{t\in[K]} t\, x_t \le \frac{KM}{N}. \tag{30}$$

Please note that (29) arises from the total number of bits across all files, which is a looser constraint than imposing that each file contains the same number of bits; similarly, (30) arises from the total number of bits across all caches, which is a looser constraint than imposing that each cache has the same size. None of these is a problem as we aim to derive a converse bound on the load at this point. This implies that that the converse bound we derive applies to the case where the total number of bits across all files and the total number of bits across all caches are constrained, but not the size each individual file or each individual cache.

The constraints in (26)-(30) provide a converse bound for the load $R^{\star}_{\text{c,u}}$ with uncoded cache placement. Since there are many inequalities in $K+1=4$ unknowns, we proceed to eliminate (x_0,x_1,x_2,x_3) in the system of inequalities in (26)-(30). By doing so, we obtain

$$\begin{aligned} R^{\star}_{\text{c,u}} &\overset{\text{by eq.(26c)}}{\geq} 3x_0+x_1+\frac{1}{3}x_2 \\ &\overset{\text{by eq.(29)}}{=} 3(1-x_1-x_2-x_3)+x_1+\frac{1}{3}x_2 \\ &= 3-2x_1-\frac{8}{3}x_2-3x_3 \\ &\overset{\text{by eq.(30)}}{\geq} 3+2(2x_2+3x_3-M)-\frac{8}{3}x_2-3x_3 \\ &= 3-2M+\frac{2}{3}x_2+3x_3 \\ &\overset{\text{by eq.(27)}}{\geq} 3-2M. \end{aligned} \tag{31}$$

Similarly, we can obtain

$$R^{\star}_{\text{c,u}} \geq -\frac{2}{3}M+\frac{5}{3}, \tag{32}$$

$$R^{\star}_{\text{c,u}} \geq -\frac{1}{3}M+1. \tag{33}$$

The maximum among the right-hand sides of (31), (32) and (33) give a piecewise linear curve with corner points: $(0,3),(1,1),(2,\frac{1}{3}),(3,0)$. Since these corner points are achieved by $R_{\text{c,u,MAN}}[t], t\in[0:3]$, in (8), we conclude that the two-phase strategy in [1] is optimal under the constraint of uncoded cache placement in this case. Note that this shows that demand vectors with distinct demands lead to the worst case load.

We are now ready to extend the reasoning in this example to a general setting, where we do not necessarily impose $K\leq N$.

C. Proof of Theorem 3

Consider a system with uncoded cache placement and a demand vector with $\min(K,N)$ distinct demanded files. We treat the delivery phase of this caching scheme as an IC problem, as described in Section II-G. We derive a converse bound on $R^{\star}_{\text{c,u}}$ using Theorem 2. A directed graph can be generated for such IC problem as described in Section II-G. We propose the following lemma to give the sets of nodes not containing a directed cycle, whose proof is in Appendix A.

Lemma 1. *Let $\mathbf{u}=(u_1,u_2,\ldots,u_{\min(K,N)})$ be a permutation of $\mathcal{C}$, where $\mathcal{C}$ is the chosen user set with different demands. A set of nodes not containing a directed cycle in the directed graph of the corresponding IC problem can be composed of sub-files*

$$\left(F_{d_{u_i},\mathcal{W}_i}:\mathcal{W}_i\subseteq[K]\backslash\{u_1,\ldots,u_i\},\ i\in[\min(K,N)]\right). \tag{34}$$

With the set of nodes not containing a directed cycle in the directed graph of the corresponding IC problem as in (34), we write the bound in (12) from Theorem 2 as

$$R^{\star}_{\text{c,u}} \geq \sum_{i\in[\min(K,N)]}\ \sum_{\mathcal{W}_i\subseteq[K]\backslash\{u_1,\ldots,u_i\}} \frac{|F_{d_{u_i},\mathcal{W}_i}|}{B}. \tag{35}$$

Note that, in the bound in (35), there are $\sum_{i\in[\min(K,N)]}\binom{K-i}{t}$ subfiles known by exactly t users whose coefficient is 1. We can also note that in general there are $N\binom{K}{t}$ subfiles known by exactly t users. By considering all sets $\mathcal{C}$ of users with $\min(K,N)$ distinct demands, and all the permutations $\mathbf{u}$ of $\mathcal{C}$, we can list all the inequalities in the form of (35) and sum them together to obtain

$$R^{\star}_{\text{c,u}} \geq \sum_{t\in[0:K]} \frac{\binom{K-1}{t}+\binom{K-2}{t}+\cdots+\binom{K-\min(K,N)}{t}}{\binom{K}{t}}\,x_t \tag{36a}$$

$$= \sum_{t\in[0:K]} \frac{\binom{K}{t+1}-\binom{K-\min(K,N)}{t+1}}{\binom{K}{t}}\,x_t, \tag{36b}$$

where from (36a) to (36b) we use the Pascal's triangle equality, where the set of coefficients $(x_0,\ldots,x_K)$ defined in (27) can be interpreted as a probability mass function (see (29)) subject to a first-moment constraint (as given in (30)).

Next, we introduce the following key Lemma, whose proof can be found in Appendix B.

Lemma 2. *Let K, N be positive integers where $K>N$. For any $q\in[K-1]$, $s_{q+1}\geq s_q$, where s_q is defined as*

$$s_q := c_q - c_{q-1}, \tag{37}$$

where c_q was defined in (16).

Lemma 2 is key in performing Fourier Motzkin elimination of x_q and x_{q-1} in (36b) for each $q\in[K]$, at the end of which we obtain the bound in (15) – see Appendix C for details.

The bound in (15) is a family of straight lines parameterized by $q\in[K]$. The lines for $q=t$ and $q=t-1$ intersect at the point in (17) because the coefficients c_t in (16) decreases monotonously in $t\in[K-1]$. This concludes the proof.

D. Remarks

a) On the equivalence of our converse bound and other bounds that appeared in the literature after we made available online our works in [4], [5]: In [13], the authors propose a genie-aided converse bound to arrive to our very same inequality in (35), where (35) was originally derived in [4], [5] by leveraging the IC acyclic converse bound. These two approaches are completely equivalent. Firstly, the IC acyclic converse bound can be proved by providing genie information to the receivers in the acyclic set. Secondly, the following steps in [13] are also the same as in our original work [4],

[5], namely, summing together all the inequalities, bounding the load by the new variables $(x_t : t \in [0:K])$ defined in (27), and eliminating the new variables to get the final converse bound. The only difference is that we use Fourier-Motzkin elimination to eliminate the new variables, while the authors in [13] treat the new variables as a probability mass function and optimized the bound over all probability mass functions (see (29)) with a given constraint on the first moment (see (30)).

b) Generalization of our converse bound to the case of average load or asymmetric settings: Our proposed converse bound trivially generalizes to different memory sizes or to different file sizes or to average load. Let the cache size of user $i \in [K]$ be $M_i B$ bits, and the length of file $j \in [N]$ be $L_j B$ bits. We have

$$R^\star_{\text{c,u}} \geq R_{\text{c,low}}, \tag{38a}$$

where $R_{\text{c,low}}$ is further lower bounded as

$$\sum_{\mathbf{d} \in [N]^K} \Pr[\mathbf{d}] R_{\mathbf{d}} \leq R_{\text{c,low}}, \text{ (for average load), or} \tag{38b}$$

$$R_{\mathbf{d}} \leq R_{\text{c,low}}, \text{ (for worst-case load),} \tag{38c}$$

where we optimize over the lengths of the subfiles subject to

$$\sum_{\mathcal{W} \subseteq [K]} |F_{j,\mathcal{W}}| = L_j B, \ \forall j \in [N], \text{ (file length),} \tag{38d}$$

$$\sum_{j \in [N]} \sum_{\mathcal{W} \subseteq [K] : u \in \mathcal{W}} |F_{j,\mathcal{W}}| \leq M_u B, \ \forall u \in [K], \text{ (cache size),} \tag{38e}$$

$$\sum_{i ß \mathcal{N}(d)} \sum_{\mathcal{W}_i \subseteq [K] \setminus \{u_1, \ldots, u_i\}} |F_{d_{u_i}, \mathcal{W}_i}| \leq R_{\mathbf{d}}, \text{ (acyclic IC bound),} \tag{38f}$$

where the demand vector $\mathbf{d} \in [N]^K$ in (38f) has $\mathcal{N}(\mathbf{d})$ distinct entries, and where the $\mathbf{u}$ is a permutation of the sub-vector of $\mathbf{d}$ with distinct entries.

The bound in (38) is a linear program with $\min\{N, K\}! \times N^K + N + K + 1$ constraints in $N2^K + 1$ variables, which becomes computationally intense to evaluate when K is large. The symmetry of the caching problem (i.e., invariance to relabeling either the files or the users) enabled us to: (i) combine together the N bounds in (38d) in the single constraint like in (29), (ii) combine together the K bounds in (38e) in the single constraint like in (30), and (iii) combine together the $\min\{N, K\}! \times N^K$ bounds in (38f) in the single constraint like in (26), without apparently any loss in optimality.

IV. Achievable Bound for Centralized Cache-aided Systems with Uncoded Cache Placement

In this section, we give an alternate proof to the one in [13] that our converse bound in Theorem 3 is indeed achievable.

A. Theorem Statement

Theorem 4. *The converse bound in Theorem 3 is achievable by the MAN uncoded cache placement and a delivery scheme based on the IC achievable scheme in Theorem 5.*

The proof of Theorem 4 is given in Section IV-B.

Theorem 5 (Novel Achievable Scheme for the General Index Coding Problem)**.** *A non-negative rate tuple* $\mathbf{R} := (R_1, \ldots, R_{N'})$ *is achievable for the IC problem* $\big((\mathcal{A}_j, \mathcal{D}_j) : j \in [K']\big)$ *defined in Section II-D if*

$$\mathbf{R} \in \bigcap_{j \in [K']} \ \bigcup_{\mathcal{K}_j : \mathcal{D}_j \subseteq \mathcal{K}_j \subseteq [N'] \setminus \mathcal{A}_j} \mathscr{R}(\mathcal{K}_j | \mathcal{A}_j, \mathcal{D}_j), \tag{39a}$$

$$\mathscr{R}(\mathcal{K} | \mathcal{A}, \mathcal{D}) := \bigcap_{\mathcal{J} : \mathcal{J} \subseteq \mathcal{K}, \mathcal{D} \cap \mathcal{J} \neq \emptyset} \left\{ \sum_{i \in \mathcal{J}} R_i < \kappa_{\mathcal{J}} \right\}, \tag{39b}$$

where in (39b) $\kappa_{\mathcal{J}}$ *is defined as*

$$\kappa_{\mathcal{J}} := I\Big((U_i : i \in \mathcal{J}); \mathbf{V} \big| (U_i : i \in \mathcal{A}_j \cup \mathcal{K}_j \setminus \mathcal{J})\Big), \tag{39c}$$

$$\mathbf{V} := \big(V_{\mathcal{P}} : \mathcal{P} \subseteq [N']\big), \tag{39d}$$

$$V_{\mathcal{P}} \text{ a function of } \big(U_i : i \in \mathcal{P}\big), \ \forall \mathcal{P} \subseteq [N'], \tag{39e}$$

for some independent auxiliary random variables $(U_i : i \in [N'])$ *and such that*

$$H\big(\mathbf{V} \big| (U_i : i \in \mathcal{A}_j)\big) < 1, \ \forall j \in [K']. \tag{39f}$$

The proof of Theorem 5 is given in Appendix D. Note that the cardinality of the auxiliary random variables U_i, $i \in [N']$, can be bounded as in [29, Section 4.2] (in particular, let $p = 1$ and $\mathcal{A}(i) = \mathcal{X}$ in [29, Theorem 4.2]), thus leading to $|\mathcal{U}_i| < |\mathcal{X}| + K'$. In addition, since $V_{\mathcal{P}}$, $\mathcal{P} \subseteq [N']$, is a function of $\big(U_i : i \in \mathcal{P}\big)$, we have $|\mathcal{V}_{\mathcal{P}}| \leq \prod_{i \in \mathcal{P}} |\mathcal{U}_i|$.

Corollary 1. *The composite (index) coding region in Theorem 1 is included in our novel Theorem 5.*

The proof of Corollary 1 is given in Appendix E.

B. Proof of Theorem 4

For centralized cache-aided systems under the constraint of uncoded cache placement, the claim of Theorem 4 is true for $N \geq K$ because the converse bound in (17) coincides with the MAN scheme in (8). For $N < K$, Theorem 5 can be used to achieve (17), as showed next.

We use the same placement phase as MAN for $M = t\frac{N}{K}$, for $t \in [0:K]$, so that the delivery phase is equivalent to an IC problem with K users in which each sub-file $F_{i,\mathcal{W}}$, for $i \in \mathcal{N}(\mathbf{d})$, $\mathcal{W} \subseteq [K]$ and $|\mathcal{W}| = t$, is an independent message, and where the desired message and side information sets are given by (13) and (14), respectively. Note that the message rates in this equivalent IC problem are identical by construction and the number of messages for the worst case-load is $N' = \min(N, K)\binom{K}{t}$.

In Theorem 5, following Example 1, we let $\mathcal{K}_j = \mathcal{D}_j$ for $j \in [K]$, we represent $F_{i,\mathcal{W}}$ as a binary vector of length $B/\binom{K}{t}$ bits (assumed to be an integer without loss of generality) and we let the corresponding random variable U be equal to the message. We also let $V_{\mathcal{P}}$ be non zero only for the linear combinations of messages sent by the MAN scheme in [1]. From (39b), for each set $\mathcal{J} \subseteq \{F_{d_k,\mathcal{W}} : k \notin \mathcal{W}\}$, we have $|\mathcal{J}| R_{\text{sym}} < |\mathcal{J}| H(U)$. With this we have $R_{\text{sym}} = H(U) = B/\binom{K}{t}$ and

$$1 = H(X) = B \frac{\binom{K}{t+1} - \binom{K - \min(N,K)}{t+1}}{\binom{K}{t}}, \tag{40}$$

so the symmetric rate is

$$R_{\text{sym}} = \frac{1}{\binom{K}{t+1} - \binom{K-\min(N,K)|}{t+1}} \tag{41}$$

Each receiver in the original caching problem is interested in recovering $\binom{K}{t}$ messages/subfiles, or one file of B bits, thus the 'sum-rate rate' delivered to each user is

$$R_{\text{sum-rate}} = \frac{\binom{K}{t}}{\binom{K}{t+1} - \binom{K-\min(N,K)}{t+1}} \tag{42}$$

The load in the caching problem is the number of transmissions (channel uses) needed to deliver one file to each user, thus the inverse of $R_{\text{sum-rate}}$ indeed corresponds to the load in (9).

C. Remarks

a) On the interpretation on the MAN scheme as source coding with side information: Our proof of Theorem 4 uses Theorem 5 and gives an interpretation of the achievable scheme proposed in [13] via the framework of *source coding with side information*. Our novel IC approach has the advantage that it applies to any IC problem, and is not limited to binary linear codes for the specific MAN placement as that of [13] when applied to the caching problem.

b) On general achievable regions for IC: The composite (index) coding region in Theorem 1 is included in our novel Theorem 5, as shown in Appendix E. The proof of Corollary 1 can be found in Appendix E. In addition, from the proof of Corollary 1, we see that the rate region achieved by composite (index) coding can be realized by linear coding.

The work in [36] improves on the composite (index) coding scheme in Theorem 1 by further rate-splitting. In Remark 2 in Appendix D we discuss how the same idea can be used to improve on Theorem 5. Finally, in Example 1 in Appendix D we give an example to show that our Theorem 5 strictly improves on [36]. It thus appears that our region in Theorem 5 is the largest known achievable region to date of the general IC based on random coding.

c) On the extension to decentralized systems: By directly extending our proposed achievable scheme in Theorem 4 and converse bound in Theorem 3 to decentralized systems, we can prove that under the constraint that each user randomly, uniformly and independently chooses MB bits of the N files to store in its local cache, the optimal load can be achieved by the novel IC achievable bound in Theorem 5; this result was shown in [4]. More precisely, for the converse part, since each user randomly, uniformly and independently chooses MB bits of the N files to store, by a Low-of-Large-Numbers type reasoning, the length of each subfile does not deviate much from its mean when the file size is large. Hence, we can use the technique proposed in Section III-C to find the largest acyclic sets of subfiles for each possible demand vector. For the achievability part, we can use the proposed delivery scheme in Section IV-B for K rounds, where in each round we transmit the subfiles known by exactly $t \in [0 : K]$ users. The details can be found in the first author's Ph.D. thesis [32].

V. CONCLUSION

In this paper we investigated the coded caching problem with uncoded cache placement by leveraging its connection to the index coding problem. We first derived a converse bound on the worst-case load of cache-aided systems under the constraint of uncoded cache placement by cleverly combining many acyclic index coding converse bounds derived by considering different demands in the caching problem. When there are more users than files, we proved that the load of the MAN scheme coincides with the proposed converse bound. In the remaining cases, our converse bound is attained by using the MAN placement phase and a delivery phase based on novel index coding scheme. The proposed novel index coding achievable scheme is based on distributed source coding and is shown to strictly improves on the well known composite (index) coding achievable bound and is, to the best our knowledge, the best random coding achievable bound for the general IC problem to date.

The present work parallels the recent results in [13]. Our main contribution compared to [13] is to build on the connection among the caching problem with uncoded cache placement and the index coding problem.

APPENDIX A
PROOF OF LEMMA 1

For a $\mathbf{u} = (u_1, u_2, \ldots, u_{\min(K,N)})$, we say that sub-files/nodes $F_{d_{u_i},\mathcal{W}_i}$, for all $\mathcal{W}_i \subseteq [K]\backslash\{u_1, \ldots, u_i\}$, are in level i. It is easy to see each node in level i only knows the sub-files $F_{j,\mathcal{W}}$ where $u_i \in \mathcal{W}$. So each node in level i knows neither the sub-files in the same level, nor the sub-files in the higher levels. As a result, in the proposed set there is no subset containing a directed cycle.

APPENDIX B
PROOF OF LEMMA 2

Recall that

$$s_q = c_q - c_{q-1}, \tag{43}$$

$$\text{and } c_q = \frac{\binom{K}{q+1} - \binom{K-\min(K,N)}{q+1}}{\binom{K}{q}}$$

$$= \frac{\binom{K-1}{q} + \cdots + \binom{K-\min(K,N)}{q}}{\binom{K}{q}}. \tag{44}$$

Focus on the first term of s_q in (43), we have

$$c_q = \frac{\binom{K-1}{q} + \cdots + \binom{K-\min(K,N)}{q}}{\binom{K}{q}} = \frac{K-q}{K} + \cdots + \frac{(K-q) \times ... \times (K-q-\min(K,N)+1)}{K \times ... \times (K-\min(K,N)+1)}. \tag{45}$$

For the second term of s_q

$$c_{q-1} = \frac{\binom{K-1}{q-1} + \cdots + \binom{K-\min(K,N)}{q-1}}{\binom{K}{q-1}} = \frac{K-q+1}{K} + \cdots + \frac{(K-q+1) \times \cdots \times (K-q-\min(K,N)+2)}{K \times \cdots \times (K-\min(K,N)+1)}. \tag{46}$$

By taking (45) and (46) into (43), we finally obtain

$$s_q = -\frac{1}{K} - \frac{2(K-q-1)}{K(K-1)} - \cdots \tag{47}$$

$$- \frac{\min(K,N)(K-q) \times \cdots \times (K-q-\min(K,N)+2)}{K \times \cdots \times (K-\min(K,N)+1)}. \tag{48}$$

Since each negative term in (48) is monotone increasing with q, it is easy to check that for any $q \in [K-1]$, $s_{q+1} \geq s_q$.

Appendix C
Proof of (15)

From (29) (i.e., the fact that $(x_0, \ldots, x_K)$ as defined in (27) can be interpreted as a probability mass function), we have

$$(c_q - qs_q)(x_q + x_{q-1}) = (c_q - qs_q)\left(1 - \sum_{i \in [0:K] \setminus \{q-1,q\}} x_i\right), \tag{49}$$

where s_q is given in (43). By the Lemma 2, for any $q \in [K-1]$, $s_{q+1} \geq s_q$. Since $s_K \leq 0$, we have $s_q \leq 0$ for all $q \in [K]$. From (30), we have

$$s_q\big((q-1)x_{q-1} + qx_q\big) \geq s_q \left(\frac{KM}{N} - \sum_{i \in [0:K] \setminus \{q-1,q\}} ix_i\right). \tag{50}$$

By summing (49) and (50) we get

$$c_{q-1}x_{q-1} + c_q x_q \geq s_q \frac{KM}{N} + c_q - s_q\, q + \sum_{i \in [0:K]: i \neq q-1,q} (-c_q + (q-i)s_q)x_i. \tag{51}$$

Next, we substitute (51) into (36b) and get

$$R^\star_{\text{c,u}} \geq \frac{s_q KM}{N} + c_q - s_q\, q + \sum_{i \in [0:K]} w_{q,i}x_i, \tag{52}$$

$$w_{q,i} := c_i - c_q + (q-i)s_q. \tag{53}$$

Note that when $i \in \{q, q-1\}$ we have $w_{q,i} = 0$. It remains to prove that for each $i \in [0:K]$ we have $w_{q,i} \geq 0$. For any $q \in [K]$ and $i \in [0:K-1]$ we have

$$w_{q,i+1} - w_{q,i} = \frac{s_{i+1}}{N} - \frac{s_q}{N}. \tag{54}$$

From Lemma 2 and (54), it can be seen that for any $q \in [K]$ and $i \in [0:K-1]$, if $i \leq q-1$, $w_{q,i+1} \leq w_{q,i}$ and if $i \geq q-1$, $w_{q,i+1} \geq w_{q,i}$. Furthermore, $w_{q,i} = 0$ for $i \in \{q, q-1\}$. Hence, for each $i \in [0:K]$, $w_{q,i} \geq 0$. As a result we have

$$R^\star_{\text{c,u}} \geq \frac{\binom{K}{q+1} - \binom{K-\min(K,N)}{q+1}}{\binom{K}{q}} + s_q\left(\frac{KM}{N} - q\right), \tag{55}$$

which proves bound given in (15).

Appendix D
Proof of Theorem 5

We introduce here a novel IC achievable scheme based on coding for the Multi-Access Channel (MAC) with correlated messages [29], Slepian-Wolf coding [30], and non-unique decoding [37]. At a very high level, the proposed scheme can be described as follows, where the terminology and notation are as in Section II-E. In the encoding stage, we generate a sequence for each message and then generate a composite function for each subset of sequences. In the decoding stage, we choose a set $\mathcal{K}_j$ such that $\mathcal{D}_j \subseteq \mathcal{K}_j$ for each user j. From all the received composite functions and the side information of user j, we let user j uniquely decode the messages in $\mathcal{D}_j$, non-uniquely decode the messages in $\mathcal{K}_j \setminus \mathcal{D}_j$, and treat the other messages as interference. We then prove that the rate region of the proposed scheme not only strictly includes the region achieved by composite (index) coding in Theorem 1 but it also strictly outperforms the improved version of Theorem 1 from [36]. Our scheme differs from Theorem 1 in the following aspects:

1) In the composite (index) coding scheme, decoder $j \in [K']$ recovers uniquely the messages in $\mathcal{K}_j$, while in our proposed scheme decoder $j \in [K']$ uniquely recovers only the desired messages indexed by $\mathcal{D}_j$ and non-uniquely the non-desired indexed by $\mathcal{K}_j \setminus \mathcal{D}_j$. So in (11b) the intersection is taken over all of $\mathcal{J} \subseteq \mathcal{K}$ while in (39b) the intersection is taken over all of $\mathcal{J} \subseteq \mathcal{K}$ such that $\mathcal{D} \cap \mathcal{J} = \emptyset$.
2) In the composite (index) coding scheme decoder $j \in [K']$ treats all the messages in $[N'] \setminus \mathcal{K}_j$ as noise and the correlation among composite indices is not considered. Thus decoder j only uses the composite indices $(W_{\mathcal{J}} : \mathcal{J} \subseteq \mathcal{K}_j \cup \mathcal{A}_j)$ to decode all the messages in $\mathcal{K}_j$. Instead, in our proposed scheme, decoder j treats all the messages in $[N'] \setminus \mathcal{K}_j$ as interference. By leveraging the correlation among all the composite functions, we let decoder j cancel the interference of $[N'] \setminus \mathcal{K}_j$.
For instance, in Example 1 at the end of this section, user 3 knows messages indexed by $\{5, 6\}$ and demands message 3, i.e., $\mathcal{A}_3 = \{5, 6\}$ and $\mathcal{D}_3 = \{3\}$. In the proposed scheme, which is proven to be optimal for this example, user 3 uses all the transmitting composite indices to recover message 3 and cancel the interference of the messages indexed by $\{1, 2, 4\}$ without decoding those messages. However, if we use composite (index) coding, user 3 can only use the composite indices $W_{\mathcal{J}}$ if and only if $\mathcal{J}$ is a subset of $\mathcal{K}_3 \cup \mathcal{A}_3$, (e.g., if we set $\mathcal{K}_3 = \mathcal{D}_3 = \{3\}$, user 3 treats all the composite indices $V_{1,3,4}$, $V_{2,4,5}$ and $V_{1,2,6}$ as noise; else if we set $\mathcal{K}_3 \supset \mathcal{D}_3$, user 3 should exactly recover all messages in $\mathcal{K}_3$ which includes some messages not demanded by user 3; in both cases, the composite coding can not achieve the converse bound). This is the main reason why composite (index) coding is not optimal in this example and why our proposed scheme outperforms composite coding.

Proof: To clarify the notations, we use different symbols for transmitted messages or known messages (nothing above), uniquely decoded ones (hat above), and non-uniquely decoded ones (check above).

a) Codebook Generation: Fix a probability mass function

$$p_{U_1,\ldots,U_{N'}}(u_1,\ldots,u_{N'}) = p_{U_1}(u_1)\times\cdots\times p_{U_{N'}}(u_{N'}), \tag{56}$$

where each random variable U_i is defined on the finite alphabet $\mathcal{U}_i$ for $i\in[N']$, and functions

$$f_{\mathcal{P}}:\prod_{i\in\mathcal{P}}\mathcal{U}_i\to\mathcal{V}_{\mathcal{P}},\quad \forall\mathcal{P}\subseteq[N'], \tag{57}$$

for some finite alphabets $\mathcal{V}_{\mathcal{P}}$ for $\mathcal{P}\subseteq[N']$.

For each $i\in[N']$, randomly and independently generate $|\mathcal{X}|^{nR_i}$ sequences $u_i^n(m_i)$ indexed by $m_i\in[|\mathcal{X}|^{nR_i}]$, each according to $\prod_{t=1}^n p_{U_i}(u_{i,t})$. For each $\mathcal{P}\subseteq[N']$, let $v_{\mathcal{P}}^n := (v_{\mathcal{P},1},\ldots,v_{\mathcal{P},n})$ and $v_{\mathcal{P},t}=f_{\mathcal{P}}\big((u_{i,t}:i\in\mathcal{P})\big)\in\mathcal{V}_{\mathcal{P}}$ where $t\in[n]$.

Randomly and independently assign an index $g\in[|\mathcal{X}|^n]$ to each collection of sequences $(v_{\mathcal{P}}^n:\mathcal{P}\subseteq[N'])$ according to a uniform probability mass function over $[|\mathcal{X}|^n]$. The sequences with the same index g are said to form bin $\mathcal{B}(g)$. We also indicate $g=\mathsf{bin}(v_{\mathcal{P}}^n:\mathcal{P}\subseteq[N'])$, the index of the bin of $v_{\mathcal{P}}^n$.

The codebook so generated is revealed to all the decoders.

b) Encoding: Given messages $(m_1,\ldots,m_{N'})$, the encoder produces $(u_1^n(m_1),\ldots,u_{N'}^n(m_{N'}))$ based on which it computes $(v_{\mathcal{P}}^n:\mathcal{P}\subseteq[N'])$ and eventually transmits $g=\mathsf{bin}(v_{\mathcal{P}}^n:\mathcal{P}\subseteq[N'])$ to all the decoders.

c) Decoding: Fix $\mathcal{K}_j$ where $\mathcal{D}_j\subseteq\mathcal{K}_j$ and $\mathcal{K}_j\cap\mathcal{A}_j=\emptyset$ for each receiver $j\in[K']$. Decoding proceeds in two steps.

Decoding Step 1: Since receiver $j\in[K']$ has messages $(m_i:i\in\mathcal{A}_j)$ as side information, it also knows $(u_i^n:i\in\mathcal{A}_j)$ and $(v_{\mathcal{P}}^n:\mathcal{P}\subseteq\mathcal{A}_j)$. Upon receiving $g=\mathsf{bin}(v_{\mathcal{P}}^n:\mathcal{P}\subseteq[N'])$, receiver $j\in[K']$ estimates the sequences $(\widehat{v}_{\mathcal{P}}^n:\mathcal{P}\subseteq[N'],\ \mathcal{P}\not\subseteq\mathcal{A}_j)$ as the unique sequences satisfying

$$\big((\widehat{v}_{\mathcal{P}}^n:\mathcal{P}\subseteq[N'],\ \mathcal{P}\not\subseteq\mathcal{A}_j),(v_{\mathcal{P}}^n:\mathcal{P}\subseteq\mathcal{A}_j)\big)\in\mathcal{B}(g); \tag{58}$$

if none or more than one are found, it picks one uniformly at random within $\mathcal{B}(g)$.

Decoding Step 2: Receiver $j\in[K']$ then uses the found $(\widehat{v}_{\mathcal{P}}^n:\mathcal{P}\subseteq[N'],\ \mathcal{P}\not\subseteq\mathcal{A}_j)$, and the side information to decode all messages in $\mathcal{K}_j$, but only those in $\mathcal{D}_j$ uniquely, that is, it finds a unique tuple $(\widehat{m}_i:i\in\mathcal{D}_j)$ and some tuple $(\check{m}_i:i\in\mathcal{K}_j\backslash\mathcal{D}_j)$ such that

$$\Big(\big(u_i^n(m_i):i\in\mathcal{A}_j\big),\ \big(u_i^n(\widehat{m}_i):i\in\mathcal{D}_j\big), \tag{59a}$$
$$\big(u_i^n(\check{m}_i):i\in\mathcal{K}_j\backslash\mathcal{D}_j\big),\ \big(\widehat{v}_{\mathcal{P}}^n:\mathcal{P}\subseteq[N'],\ \mathcal{P}\not\subseteq\mathcal{A}_j\big), \tag{59b}$$
$$\big(v_{\mathcal{P}}^n:\mathcal{P}\subseteq\mathcal{A}_j\big)\Big) \tag{59c}$$
$$\in T_\varepsilon^{(n)}\Big(\big(U_i:i\in\mathcal{A}_j\cup\mathcal{K}_j\big),\big(V_{\mathcal{P}}:\mathcal{P}\subseteq[N']\big)\Big); \tag{59d}$$

if none or more than one $(\widehat{m}_i:i\in\mathcal{D}_j)$ are found, it picks one uniformly at random.

d) Error Analysis: For each decoder $j\in[K']$ and $\mathcal{J}\subseteq\mathcal{K}_j$ where $\mathcal{J}\cap\mathcal{D}_j\neq\emptyset$, we define the error events in (60) at the top of the next page.

For decoder j, the probability of error at decoder j denoted by $\Pr(\mathcal{E}(j))$ can be upper bounded by

$$\Pr\big(\mathcal{E}(j)\big)\le\Pr(\mathcal{E}_1) \tag{61a}$$
$$+\Pr\big(\mathcal{E}_1^c\cap\mathcal{E}_{2,j}|\mathcal{B}(1)\big) \tag{61b}$$
$$+\sum_{\mathcal{J}\subseteq\mathcal{K}_j:\mathcal{J}\cap\mathcal{D}_j\neq\emptyset}\Pr\big(\mathcal{E}_{j,\mathcal{J}}\cap\mathcal{E}_1^c\cap\mathcal{E}_{1,j}^c\big). \tag{61c}$$

We now bound each term in (61). By LLN, the term in (61a) vanishes as $n\to\infty$. Next, for the term in (61b) we have

$$\Pr\big(\mathcal{E}_1^c\cap\mathcal{E}_{2,j}|\mathcal{B}(1)\big)\le\sum_{(u_i^n:i\in[N'])}\Pr\Big\{U_i^n=u_i^n,i\in[N']$$
$$\Big|\Big(f_{\mathcal{P}}\big((U_i^n:i\in\mathcal{P})\big):\mathcal{P}\subseteq[N']\Big)\in\mathcal{B}(1)\Big\}q_{(u_i^n:i\in[N'])} \tag{62}$$

where

$$\begin{aligned}&q_{(u_i^n:i\in[N'])}:=\\&\Pr\Big\{\big((\widehat{v}_{\mathcal{P}}^n:\mathcal{P}\subseteq[N']\text{ and }\mathcal{P}\not\subseteq\mathcal{A}_j),(x_{\mathcal{P}}^n:\mathcal{P}\subseteq\mathcal{A}_j)\big)\\&\in\mathcal{B}(1)\text{ for some }(\widehat{v}_{\mathcal{P}}^n:\mathcal{P}\subseteq[N']\text{ and }\mathcal{P}\not\subseteq\mathcal{A}_j)\in\\&\mathcal{G}_{(u_i^n:i\in[N'])}\Big|\Big(f_{\mathcal{P}}\big((U_i^n:i\in\mathcal{P})\big):\mathcal{P}\subseteq[N']\Big)\in\mathcal{B}(1),\\&U_i^n=u_i^n\text{ where }i\in[N']\Big\};\\&\mathcal{G}_{(u_i^n:i\in[N'])}:=\Big\{(\widehat{v}_{\mathcal{P}}^n:\mathcal{P}\subseteq[N']\text{ and }\mathcal{P}\not\subseteq\mathcal{A}_j)\neq\\&\Big(f_{\mathcal{P}}\big((u_i^n:i\in\mathcal{P})\big):\mathcal{P}\subseteq[N']\text{ and }\mathcal{P}\not\subseteq\mathcal{A}_j\Big):\\&(\widehat{v}_{\mathcal{P}}^n:\mathcal{P}\subseteq[N']\text{ and }\mathcal{P}\not\subseteq\mathcal{A}_j)\in\\&T_\varepsilon^{(n)}\big((V_{\mathcal{P}}:\mathcal{P}\subseteq[N']\text{ and }\mathcal{P}\not\subseteq\mathcal{A}_j)|(u_i^n:i\in\mathcal{A}_j)\big)\Big\}.\end{aligned} \tag{63}$$

We then focus on $q_{(u_i^n:i\in[N'])}$ to obtain

$$\begin{aligned}&q_{(u_i^n:i\in[N'])}\le\sum_{\substack{(\widehat{v}_{\mathcal{P}}^n:\mathcal{P}\subseteq[N']\text{ and }\mathcal{P}\not\subseteq\mathcal{A}_j)\in\\T_\varepsilon^{(n)}\left((X_{\mathcal{P}}:\mathcal{P}\subseteq[N']\text{ and }\mathcal{P}\not\subseteq\mathcal{A}_j)|(u_i^n:i\in\mathcal{A}_j)\right)}}\\&\Pr\Big\{\big((\widehat{v}_{\mathcal{P}}^n:\mathcal{P}\subseteq[N'],\mathcal{P}\not\subseteq\mathcal{A}_j),(x_{\mathcal{P}}^n:\mathcal{P}\subseteq\mathcal{A}_j)\big)\in\mathcal{B}(1)\Big|\\&\Big(f_{\mathcal{P}}\big((U_i^n:i\in\mathcal{P})\big):\mathcal{P}\subseteq[N']\Big)\in\mathcal{B}(1),U_i^n=u_i^n\forall i\in[N']\Big\}\\&\le|\mathcal{X}|^{n\left[H\left((V_{\mathcal{P}}:\mathcal{P}\subseteq[N'],\mathcal{P}\not\subseteq\mathcal{A}_j)|(U_i:i\in\mathcal{A}_j)\right)\right]}|\mathcal{X}|^{-n}.\end{aligned} \tag{64}$$

From (62) and (64) we can see that the term in (61a) vanishes provided that

$$H\big((V_{\mathcal{P}}:\mathcal{P}\subseteq[N']\text{ and }\mathcal{P}\not\subseteq\mathcal{A}_j)|(U_i:i\in\mathcal{A}_j)\big)<1. \tag{65}$$

Finally, for the term in (61c) also vanishes, provided that (by the packing lemma [31, Lemma 3.1])

$$\sum_{i\in\mathcal{J}}R_i<I\big((U_i:i\in\mathcal{J});(V_{\mathcal{P}}:\mathcal{P}\subseteq[N']\text{ and} \tag{66}$$
$$\mathcal{P}\not\subseteq\mathcal{K}_j\cup\mathcal{A}_j\setminus\mathcal{J}),(U_i:i\in\mathcal{K}_j\cup\mathcal{A}_j\setminus\mathcal{J})\big) \tag{67}$$
$$=I\big((U_i:\in\mathcal{J});(V_{\mathcal{P}}:\mathcal{P}\subseteq[N']\text{ and }\mathcal{P}\not\subseteq\mathcal{K}_j\cup\mathcal{A}_j\setminus\mathcal{J})|(U_i:i\in\mathcal{K}_j\cup\mathcal{A}_j\setminus\mathcal{J})\big). \tag{68}$$

$$\mathcal{E}_1 := \big\{\big((U_i^n(M_i) : i \in [N']), (V_{\mathcal{P}}^n : \mathcal{P} \subseteq [N'])\big) \notin T_\varepsilon^{(n)}\big((U_i : i \in [N']), (V_{\mathcal{P}}^n : \mathcal{P} \subseteq [N'])\big)\big\}, \tag{60a}$$

$$\begin{aligned}\mathcal{E}_{2,j} := \big\{&\text{there exists } (\widehat{v}_{\mathcal{P}}^n : \mathcal{P} \subseteq [N'] \text{ and } \mathcal{P} \nsubseteq \mathcal{A}_j) \in T_\varepsilon^{(n)}\big((V_{\mathcal{P}} : \mathcal{P} \subseteq [N'] \text{ and } \mathcal{P} \nsubseteq \mathcal{A}_j)|(U_i^n : i \in \mathcal{A}_j)\big)\\ &\text{such that } \big((\widehat{v}_{\mathcal{P}}^n : \mathcal{P} \subseteq [N'] \text{ and } \mathcal{P} \nsubseteq \mathcal{A}_j), (V_{\mathcal{P}}^n : \mathcal{P} \subseteq \mathcal{A}_j)\big) \in \mathcal{B}(G) \text{ and } (\widehat{v}_{\mathcal{P}}^n : \mathcal{P} \subseteq [N']\\ &\text{and } \mathcal{P} \nsubseteq \mathcal{A}_j) \neq (V_{\mathcal{P}}^n : \mathcal{P} \subseteq [N'] \text{ and } \mathcal{P} \nsubseteq \mathcal{A}_j)\big\}, \text{ where } G \text{ is the random index of } g,\end{aligned} \tag{60b}$$

$$\begin{aligned}\mathcal{E}_{j,\mathcal{J}} := \big\{&\text{there exists } \widehat{m}_i \neq M_i \text{ where } i \in \mathcal{J} \text{ such that } \big((U_i^n(M_i) : i \in \mathcal{K}_j \cup \mathcal{A}_j \setminus \mathcal{J}), (U_i^n(\widehat{m}_i) : i \in \mathcal{J}),\\ &(V_{\mathcal{P}}^n : \mathcal{P} \subseteq [N'])\big) \in T_\varepsilon^{(n)}\big((U_i : i \in \mathcal{K}_j \cup \mathcal{A}_j), (V_{\mathcal{P}} : \mathcal{P} \subseteq [N'])\big)\big\}.\end{aligned} \tag{60c}$$

The inequalities in (65) and in (66) are strict; however, by the same argument in the composite-coding achievable region in [28, Proposition 6.11] they can be relaxed to be nonstrict. ■

Remark 2. *The work in [36] improves on the composite (index) coding scheme in Theorem 1 by splitting each message into non-overlapping and independent sub-message*

$$M_i = \big(M_i\big((\mathcal{K}_1, \ldots, \mathcal{K}_{K'})\big) : \mathcal{K}_k \subseteq [N'] \setminus \mathcal{A}_k, \ \text{for } k \in [K']\big).$$

Composite (index) coding is then used to transmit each group of sub-messages, $\{M_i\big((\mathcal{K}_1, \ldots, \mathcal{K}_{K'})\big) : i \in [N']\}$. A drawback of this scheme is that the number of auxiliary variable increases exponentially with the number of users.

We could also use this message-splitting idea to improve our achievable region in Theorem 5. If we did so, then the improved version of Theorem 5 would include the improved version of Theorem 1. This message-splitting improvement is quite straightforward and not pursued here. We note that the improvement in performance comes at the expense of a much heaver notation, and–more importantly–a much larger computation burden to evaluate a regions that is already combinatorial in nature.

We would like to stress that our main objective here is to propose a general IC achievable scheme that can achieve the converse bound for the caching problem under the constraint of uncoded cache placement, and at the same time improves on the composite (index) coding. To the best of our knowledge, such general scheme does not exist in the literature. Corollary 1 and of Theorem 5 show that the current achievable bound in Theorem 5 is sufficient to achieve our objective. In addition, in some special cases such as in the following Example 1, we show that even if we do not use rate splitting, our proposed achievable bound is strictly better than the the message-split region in [36, SectionIII-B].

Example 1. Consider a multiple unicast IC problem with $K' = 6$ messages and with

$$\begin{aligned}\mathcal{D}_1 &= \{1\}, \quad \mathcal{A}_1 = \{3,4\},\\ \mathcal{D}_2 &= \{2\}, \quad \mathcal{A}_2 = \{4,5\},\\ \mathcal{D}_3 &= \{3\}, \quad \mathcal{A}_3 = \{5,6\},\\ \mathcal{D}_4 &= \{4\}, \quad \mathcal{A}_4 = \{2,3,6\},\\ \mathcal{D}_5 &= \{5\}, \quad \mathcal{A}_5 = \{1,4,6\},\\ \mathcal{D}_6 &= \{6\}, \quad \mathcal{A}_6 = \{1,2\}.\end{aligned}$$

Composite (Index) Coding Achievable Bound: In [36, Example 1] the authors showed that the largest symmetric rate with the composite (index) coding achievable bound in Theorem 1 for this problem is $R_{\text{sym,cc}} = 0.2963$. It the same paper, the authors proposed an extension of the composite (index) coding idea (see [36, Section III.B]) and showed that this extended scheme for this problem gives $R_{\text{sym,cc,enhanced}} = 0.2987$.

Converse: Give message M_5 as additional side information to receiver 1 so that the new side information set satisfied $\{3,4,5\} \subseteq \mathcal{A}_1$. With this side information, in addition to message 1, receiver 1 can decode message 2 and then message 6 for a total of 3 messages. Thus

$$3R_{\text{sym}} \leq \lim_{n\to\infty} \frac{1}{n} H(X^n) \leq 1, \tag{69}$$

where R_{sym} denotes the symmetric rate. Next we show that $R_{\text{sym}} \leq 1/3$ is tight. *This shows the strict sub-optimality of composite (index) coding and its message-split extension.*

Achievability: It is not difficult to see that all users can be satisfied by the transmission of the three coded messages $X = (M_1 \oplus M_3 \oplus M_4, \ M_2 \oplus M_4 \oplus M_5, \ M_1 \oplus M_2 \oplus M_6)$.

We now map this scheme into a choice of auxiliary random variables in our novel IC scheme in Theorem 5. Let $\mathcal{K}_j = \mathcal{D}_j$ for $j \in [6]$, and

$$\begin{aligned}&U_1 = M_1, \ U_2 = M_2, \cdots, U_6 = M_6,\\ &\text{for all } \mathcal{P} \subseteq [6] \text{ set } V_{\mathcal{P}} = 0 \text{ except for the following}\\ &V_{\{1,3,4\}} = U_1 \oplus U_3 \oplus U_4,\\ &V_{\{2,4,5\}} = U_2 \oplus U_4 \oplus U_5,\\ &V_{\{1,2,6\}} = U_1 \oplus U_2 \oplus U_6,\end{aligned}$$

Hence, $\mathbf{V} = (V_{\{1,3,4\}}, V_{\{2,4,5\}}, V_{\{1,2,6\}})$. From (39b), we have that for example the rate bound corresponding to receiver 5 is

$$\begin{aligned}&R_{\text{sym}} < I(U_5; \mathbf{V}|U_1, U_4, U_6)\\ &= I(U_5; U_3, U_2 \oplus U_5, U_2) = I(U_5; U_2, U_3, U_5) = I(U_5; U_5)\\ &= H(U_5) = 1/3,\end{aligned}$$

and similarly for all the other users. As a result, any $R_{\text{sym}} < 1/3$ is achievable by the proposed scheme based on random coding argument; by repeating the same argument with random linear codes, $R_{\text{sym}} \leq 1/3$ is achievable and coincides with the converse bound. □

APPENDIX E
PROOF OF COROLLARY 1

In general, for a set $\mathcal{B} \subseteq [N']$ and for the auxiliary random variables as defined in Theorem 5, we have

$$\begin{aligned}
&H\big((V_{\mathcal{P}} : \mathcal{P} \subseteq [N'])\big|(U_i : i \in \mathcal{B})\big) \\
&\leq H\big((V_{\mathcal{P}} : \mathcal{P} \subseteq [N'], \mathcal{P} \not\subseteq \mathcal{B})\big) \\
&\leq \sum_{\mathcal{P}:\mathcal{P}\subseteq[N'],\mathcal{P}\not\subseteq\mathcal{B}} H\big(V_{\mathcal{P}}\big) \\
&\leq \sum_{\mathcal{P}:\mathcal{P}\subseteq[N'],\mathcal{P}\not\subseteq\mathcal{B}} S_{\mathcal{P}}, \quad \text{where } S_{\mathcal{P}} = \log_{|\mathcal{X}|}(|\mathcal{V}_{\mathcal{P}}|). \qquad (70)
\end{aligned}$$

In the following, we assume $|\mathcal{X}|$ is large enough such that $\{\log_2(|\mathcal{X}|)S_{\mathcal{P}} : \mathcal{P} \subseteq [N']\}$ are integers; this is so because the cardinality of the input alphabet does not affect the capacity region as argued in Remark 1. We choose the auxiliary random variables $(U_i : i \in [N'])$ and $(V_{\mathcal{P}} : \mathcal{P} \subseteq [N'])$ such that all the inequality leading to (70) holds with equality for any $\mathcal{B} \subseteq [N']$, that is, we construct random variables $(V_{\mathcal{P}} : \mathcal{P} \subseteq [N'])$ that are independent and uniformly distributed, where the alphabet of $V_{\mathcal{P}}$ has support of size $|\mathcal{V}_{\mathcal{P}}| = |\mathcal{X}|^{S_{\mathcal{P}}} = 2^{\log_2(|\mathcal{X}|)S_{\mathcal{P}}}$. With this choice of auxiliary random variables we show that Theorem 5 reduces to Theorem 1.

More precisely, let U_i, for $i \in [N']$, be an independent and equally likely binary vector of length L_i. For all $\mathcal{P} \subseteq [N']$, let $V_{\mathcal{P}}$ be a binary vector of length $\log_2(|\mathcal{X}|)S_{\mathcal{P}}$ obtained as a linear code for the collection of bits in $(U_i : i \in \mathcal{P})$. We let $L_i = \sum_{\mathcal{P}\subseteq[N']:i\in\mathcal{P}} \log_2(|\mathcal{X}|)S_{\mathcal{P}}$ for all $i \in [N']$, and divide the L_i bits in U_i into $2^{N'-1}$ non-overlapping parts, where $U_i = (U_{i,\mathcal{P}} : \mathcal{P} \subseteq [N'] : i \in \mathcal{P})$ and $|U_{i,\mathcal{P}}| = \log_2(|\mathcal{X}|)S_{\mathcal{P}}$. For each $\mathcal{P} \subseteq [N']$ where $|\mathcal{P}| > 0$, we let

$$V_{\mathcal{P}} = \bigoplus_{i\in\mathcal{P}} U_{i,\mathcal{P}}. \qquad (71)$$

Now let us focus on a set $\mathcal{B} \subseteq [N']$. We have

$$\begin{aligned}
&H\big((V_{\mathcal{P}} : \mathcal{P} \subseteq [N'])\big|(U_i : i \in \mathcal{B})\big) \\
&= H\big((V_{\mathcal{P}} : \mathcal{P} \subseteq [N'], \mathcal{P} \not\subseteq \mathcal{B}\big|(U_i : i \in \mathcal{B})\big) \\
&= \sum_{j\in\left[2^{N'}-2^{|\mathcal{B}|}\right]} H\big(V_{\mathcal{P}(\mathcal{B},j)}\big|(U_i : i \in \mathcal{B}), V_{\mathcal{P}(\mathcal{B},1)}, \ldots, V_{\mathcal{P}(\mathcal{B},j-1)}\big), \qquad (72)
\end{aligned}$$

where we randomly order the sets $\mathcal{P} \subseteq [N']$ where $\mathcal{P} \not\subseteq \mathcal{B}$, and we denote them by $\mathcal{P}(\mathcal{B},1), \mathcal{P}(\mathcal{B},2), \ldots, \mathcal{P}(\mathcal{B}, 2^{N'}-2^{|\mathcal{B}|})$. We focus on one $j \in \left[2^{N'} - 2^{|\mathcal{B}|}\right]$,

$$\begin{aligned}
&H\big(V_{\mathcal{P}(\mathcal{B},j)}\big|(U_i : i \in \mathcal{B}), V_{\mathcal{P}(\mathcal{B},1)}, \ldots, V_{\mathcal{P}(\mathcal{B},j-1)}\big) \\
&\geq H\big(V_{\mathcal{P}(\mathcal{B},j)}\big|(U_i : i \in \mathcal{B}), (U_{i,\mathcal{P}(\mathcal{B},1)} : i \in \mathcal{P}(\mathcal{B},1)), \ldots, \\
&(U_{i,\mathcal{P}(\mathcal{B},j-1)} : i \in \mathcal{P}(\mathcal{B}, j-1))\big) \\
&= H\big(V_{\mathcal{P}(\mathcal{B},j)}\big|(U_i : i \in \mathcal{B})\big) \qquad (73a) \\
&= H\big(V_{\mathcal{P}(\mathcal{B},j)}\big) \qquad (73b) \\
&= \log_{|\mathcal{X}|}\left(2^{\log_2(|\mathcal{X}|)S_{\mathcal{P}(\mathcal{B},j)}}\right) \qquad (73c) \\
&= S_{\mathcal{P}(\mathcal{B},j)}, \qquad (73d)
\end{aligned}$$

where (73a) follows from the fact that $V_{\mathcal{P}(\mathcal{B},j)}$ is independent of $(U_{i,\mathcal{P}(\mathcal{B},1)} : i \in \mathcal{P}(\mathcal{B},1)), \ldots, (U_{i,\mathcal{P}(\mathcal{B},j-1)} : i \in \mathcal{P}(\mathcal{B}, j-1))\big)$, (73b) from that $\mathcal{P}(\mathcal{B},j) \not\subseteq \mathcal{B}$, (73c) from that the bits in $V_{\mathcal{P}(\mathcal{B},j)}$ are i.i.d.. From (73b), (72) and (70), by our construction we can achieve

$$H\big((V_{\mathcal{P}} : \mathcal{P} \subseteq [N'])\big|(U_i : i \in \mathcal{B})\big) = \sum_{\mathcal{P}:\mathcal{P}\subseteq[N'],\mathcal{P}\not\subseteq\mathcal{B}} S_{\mathcal{P}}. \qquad (74)$$

As a result, we have that the bound in (39f) reduces to the one in (11d) by using (74) with $\mathcal{B} = \mathcal{A}_j$, and that the bound in (39c) reduces to the one in (11c) by using (74) twice, once with $\mathcal{B} = \mathcal{A} \cup \mathcal{K} \setminus \mathcal{J}$ and once with $\mathcal{B} = \mathcal{A} \cup \mathcal{K}$, which is so because

$$\begin{aligned}
\kappa_{\mathcal{J}} &= \sum_{\mathcal{P}:\mathcal{P}\subseteq[N']:\mathcal{P}\not\subseteq(\mathcal{A}\cup\mathcal{K}\setminus\mathcal{J})} S_{\mathcal{P}} - \sum_{\mathcal{P}:\mathcal{P}\subseteq[N']:\mathcal{P}\not\subseteq(\mathcal{A}\cup\mathcal{K})} S_{\mathcal{P}} \qquad (75) \\
&= \sum_{\mathcal{P}:\mathcal{P}\subseteq\mathcal{A}\cup\mathcal{K}:\mathcal{P}\cap\mathcal{J}\neq\emptyset} S_{\mathcal{P}}. \qquad (76)
\end{aligned}$$

This concludes the proof.

REFERENCES

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

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

[3] M. Ji, A. M. Tulino, J. Llorca, and G. Caire, "Order-optimal rate of caching and coded multicasting with random demands," *IEEE Trans. Infor. Theory*, vol. 63, pp. 3923–3949, Jun. 2017.

[4] K. Wan, D. Tuninetti, and P. Piantanida, "On the optimality of uncoded cache placement," *in 2016 IEEE Infor. Theory Workshop (first publicly available in Nov. 2015, at arXiv:1511.02256)*, 2016.

[5] ——, "On caching with more users than files," *in 2016 IEEE Int. Symp. Inf. Theory (first publicly available in Jan. 2016, at arXiv:1601.06383)*, 2016.

[6] C. Yapar, K. Wan, R. F. Schaefer, and G. Caire, "On the optimality of d2d coded caching with uncoded cache placement and one-shot delivery," *available at arXiv:1901.05921*, Jan. 2019.

[7] A. M. Ibrahim, A. A. Zewail, and A. Yener, "Device-to-device coded caching with heterogeneous cache sizes," *in IEEE Int. Conf. Communications (ICC)*, May 2018.

[8] K. Wan, M. Ji, P. Piantanida, and D. Tuninetti, "Novel outer bounds for combination networks with end-user-caches," *in IEEE Inf. Theory Workshop*, Nov. 2017.

[9] E. Parrinello, A. Unsal, and P. Elia, "Coded caching with shared caches: Fundamental limits with uncoded prefetching," *arXiv:1809.09422*, Sep. 2018.

[10] M. A. Attia and R. Tandon, "Approximately optimal distributed data shuffling," *in IEEE Int. Symp. Inf. Theory*, Jul. 2018.

[11] C.-Y. Wang, S. H. Lim, and M. Gastpar, "A new converse bound for coded caching," *in Inf. Theory and Applications Workshop (ITA)*, Jan. 2016.

[12] M. M. Amiri, Q. Yang, and D. Gunduz, "Coded caching for a large number of users," *in IEEE Inf. Theory Workshop*, Sep. 2016.

[13] Q. Yu, M. A. Maddah-Ali, and A. Salman, "The exact rate-memory tradeoff for caching with uncoded prefetching," *in 2017 IEEE Int. Symp. Inf. Theory (first publicly available in Sep. 2016, at arXiv:1609.07817)*, 2017.

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

[15] Z. Chen, "Fundamental limits of caching: Improved bounds for small buffer users," *IET Communications*, vol. 10, no. 17, Nov. 2016.

[16] S. Sahraei and M. Gastpar, "K users caching two files: An improved achievable rate," *Annual Conference on Information Science and Systems (CISS)*, Mar. 2016.

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

[18] J. Gomez-Vilardebo, "Fundamental limits of caching: Improved bounds with coded prefetching," *available at arXiv:1612.09071*, Jan 2017.

[19] H. Ghasemi and A. Ramamoorthy, "Further results on lower bounds for coded caching," *in IEEE Int. Symp. Inf. Theory*, pp. 2319 – 2323, Jul. 2016.

[20] N. Ajaykrishnan, N. S. Prem, V. M. Prabhakaran, and R. Vaze, "Critical database size for effective caching," *National Conference on Communications (NCC)*, Feb. 2015.

[21] A. Sengupta, R. Tandon, and T. C. Clancy, "Improved approximation of storage-rate tradeoff for caching via new outer bounds," *in IEEE Int. Symp. Inf. Theory*, June. 2015.

[22] C. Tian, "Symmetry, demand types and outer bounds in caching systems," *in IEEE Int. Symp. Inf. Theory*, pp. 825–829, Jul. 2016.

[23] C.-Y. Wang, S. S. Bidokhti, and M. Wigger, "Improved converses and gap-results for coded caching," *IEEE Trans. Infor. Theory*, vol. 64, no. 11, pp. 7051–7062, Jul. 2018.

[24] T. M. Cover and J. A. Thomas, "Elements of information theory," *2nd ed. New York, NY, USA: Wiley-interscience*, 2006.

[25] S. S. Bidokhti, M. Wigger, and A. Yener, "Benefits of cache assignment on degraded broadcast channels," *in IEEE Int. Symp. Inf. Theory*, Jun. 2017.

[26] Q. Yu, M. A. Maddah-Ali, and A. Salman, "Characterizing the rate-memory tradeoff in cache networks within a factor of 2," *in IEEE Int. Symp. Inf. Theory*, Jun. 2017.

[27] K. Wan, M. Ji, P. Piantanida, and D. Tuninetti, "Novel outer bounds with uncoded cache placement for combination networks with end-user-caches," *in IEEE Inf. Theory Workshop (ITW)*, Nov. 2017.

[28] F. Arbabjolfaei, B. Bandemer, Y.-H. Kim, E. Sasoglu, and L. Wang, "On the capacity region for index coding," *in IEEE Int. Symp. Inf. Theory*, Jul. 2013.

[29] T. S. Han, "The capacity region of general multiple-access channel with certain correlated sources," *Inf. Contr.*, vol. 40, no. 1, pp. 37–60, 1979.

[30] D. Slepian and J. K. Wolf, "Noiseless coding of correlated information sources," *IEEE Trans. Infor. Theory*, vol. 19, no. 4, pp. 471–480, Jul. 1973.

[31] A. E. Gamal and Y.-H. Kim, "Network information theory," *Cambridge University Press*, 2011.

[32] K. Wan, "Fundamental limits of cache-aided shared-link broadcast networks and combination networks," *Ph.D. thesis*, Jun. 2018, available at https://tel.archives-ouvertes.fr/tel-01842269/.

[33] Y. Birk and T. Kol, "Informed source coding on demand (iscod) over broadcast channels," *in Proc. IEEE Conf. Comput. Commun.*, p. 1257–1264, 1998.

[34] F. Arbabjolfaei and Y.-H. Kim, "Fundamentals of index coding," *Foundations and Trends in Communications and Information Theory*, vol. 14, no. 3-4, pp. 163–346, Oct. 2018.

[35] A. Blasiak, R. Kleinberg, and E. Lubetzky, "Lexicographic products and the power of non-linear network coding," *in IEEE 52nd Annu. Symp. Found. Comput. Sci.*, pp. 609–618, Oct. 2011.

[36] Y. Liu, P. Sadeghi, F. Arbabjolfaei, and Y.-H. Kim, "On the capacity for distributed index coding," *in IEEE Int. Symp. Inf. Theory*, Jun. 2017.

[37] B. Bandemer, A. E. Gamal, and Y. H. Kim, "Simultaneous nonunique decoding is rate-optimal," *in Proc. 50th Allerton Conf.*, pp. 9–16, Oct. 2012.

Kai Wan (S '15 – M '18) received the M.Sc. and Ph.D. degrees in Communications from Université Paris Sud-CentraleSupélec, France, in 2014 and 2018. He is currently a post-doctoral researcher with the Communications and Information Theory Chair (CommIT) at Technische Universität Berlin, Berlin, Germany. His research interests include coded caching, index coding, distributed storage, wireless communications, and distributed computing.

Daniela Tuninetti (M '98 – SM '13) is currently a Professor within the Department of Electrical and Computer Engineering at the University of Illinois at Chicago (UIC), which she joined in 2005. Dr. Tuninetti got her Ph.D. in Electrical Engineering in 2002 from ENST/Télécom ParisTech (Paris, France, with work done at the Eurecom Institute in Sophia Antipolis, France), and she was a postdoctoral research associate at the School of Communication and Computer Science at the Swiss Federal Institute of Technology in Lausanne (EPFL, Lausanne, Switzerland) from 2002 to 2004. Dr. Tuninetti is a recipient of a best paper award at the European Wireless Conference in 2002, of an NSF CAREER award in 2007, and named University of Illinois Scholar in 2015. Dr. Tuninetti was the editor-in-chief of the IEEE Information Theory Society Newsletter from 2006 to 2008, an editor for IEEE COMMUNICATION LETTERS from 2006 to 2009, for IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS from 2011 to 2014; and for IEEE TRANSACTIONS ON INFORMATION THEORY from 2014 to 2017. She is currently a distinguished lecturer for the Information Theory society. Dr. Tuninetti's research interests are in the ultimate performance limits of wireless interference networks (with special emphasis on cognition and user cooperation), coexistence between radar and communication systems, multi-relay networks, content-type coding, cache-aided systems and distributed private coded computing.

Pablo Piantanida (SM '16) received both B.Sc. in Electrical Engineering and the M.Sc (with honors) from the University of Buenos Aires (Argentina) in 2003, and the Ph.D. from Université Paris-Sud (Orsay, France) in 2007. Since October 2007 he has joined the Laboratoire des Signaux et Systèmes (L2S), at CentraleSupélec together with CNRS (UMR 8506) and Université Paris-Sud, as an Associate Professor of Network Information Theory. He is currently associated with Montreal Institute for Learning Algorithms (Mila) at Université de Montréal, Quebec, Canada. He is an IEEE Senior Member and serves as Associate Editor for IEEE Transactions on Information Forensics and Security. He served as General Co-Chair of the 2019 IEEE International Symposium on Information Theory (ISIT). His research interests lie broadly in information theory and its interactions with other fields, including multi-terminal information theory, Shannon theory, machine learning and representation learning, statistical inference, cooperative communications, communication mechanisms for security and privacy.