On the Fundamental Limit of Coded Caching Systems with a Single Demand Type

Shuo Shao
Dept. of Electrical Engr.
Shanghai Jiao Tong University
shuoshao@sjtu.edu.cn

Jesús Gómez-Vilardebó
Centre Tecnològic de Telecomunicacions
de Catalunya (CTTC/CERCA)
jesus.gomez@cttc.cat

Kai Zhang and Chao Tian
Dept. of Electrical and Computer Engr.
Texas A&M University
{kaizhang,chao.tian}@tamu.edu

***Abstract*—Caching is a technique to reduce the communication load in peak hours by prefetching contents during off-peak hours. Recently Maddah-Ali and Niesen introduced an information theoretic framework for coded caching, and showed that significant improvement can be obtained compared to uncoded caching. Considerable efforts have been devoted to identify the precise information theoretic fundamental limit of such systems, however the difficulty of this task has also become clear. One of the reasons for this difficulty is that the original coded caching setting allows multiple demand types during delivery, which in fact introduces tension in the coding strategy to accommodate all of them. In this paper, we seek to develop a better understanding of the fundamental limit of coded caching by investigating single demand type systems. We first show that in the canonical three-user three-file systems, such single demand type systems already provide important insights. Motivated by these findings, we focus on systems where the number of users and the number of files are the same, and the demand type is when all files are being requested. A novel coding scheme is proposed, which provides several optimal memory-transmission operating points. Outer bounds for this class of systems are also considered, and their relation with existing bounds is discussed.**

I. Introduction

Caching is a technique to alleviate communication load during peak hours by prefetching certain contents to the memory of the end users during off-peak hours. Recently, Maddah-Ali and Niesen [1] proposed an information theoretic framework for caching, and showed that coded caching can achieve significant improvement over uncoded caching. This caching system, with N files and K users, operates in two stages: during the prefetching stage, each user fills the cache memory of size M with information on the files, and during the delivery stage, the users reveal their requests, and the central server broadcasts common information of size R to all the users, which can be used jointly with the cached contents to fulfill the requests.

The optimal tradeoff between M and R is of fundamental importance in this setting, the characterization of which has attracted significant research effort. Schemes for both uncoded prefetching and coded prefetching have been proposed [1]–[9], and various outer bounds have also been discovered [11]–[14]. Nevertheless, except a few special cases, the fundamental limit of coded caching systems still remains unknown.

In the placement phase, each user has no prior knowledge of the demands in the delivery phase, and the prefetched contents need to be properly designed to accommodate all possible demand vectors. In a recent work [10] (see also [11]), the notion of demand type was introduced to classify the demand vectors, which lead to simplifications in a computer-aided investigation of the outer bounds. From this perspective, the original setting [1] in fact allows fully mixed demand types, and it appears that one reason for the afore-mentioned difficult is the tension among the coding requirements to accommodate these demand types. Thus a natural question is how different demand types impact this optimal (M, R) tradeoff.

To develop better understanding of this issue, in this work we consider single-demand type systems, where during the placement phase, the users and server know a priori that the demand vector in the delivery phase must be of a given demand type. Clearly, single-demand type systems have a more relaxed coding requirement than the original setting, however, it is still highly nontrivial since each single demand type allows a rich set of possible demand vectors. Because of the relaxed coding requirement, schemes for the fully mixed demand type system will also be valid for a single demand type system, however, an outer bound for the fully mixed demand type systems may not hold for a single demand type system.

Our main contribution in this paper is as follows. Firstly, in Section III we collect the best known inner bounds and outer bounds for both fully mixed demand type systems and single demand type systems in the literature for the canonical $(N, K) = (3, 3)$ system, some of which are from very recent developments [6], [8] in the area. This exercise reveals that although the demand type where all files are requested may pose a significant challenge in terms of characterizing the fundamental limit, codes designed for this demand type can in fact achieve (M, R) pairs that are strictly impossible for another demand type. This is contrary to popular belief that such a demand type is the "worst case", and also confirms that there is indeed a tension for codes designed for different demand types, and a fully mixed system would need to balance such conflicting interests. Next, in Section IV we focus on the case $N = K$ and for the demand type where all files

The work of Shuo Shao is supported by NSFC-61872149. The work of J. Gómez-Vilardebó was supported in part by the Catalan Government under Grant SGR2017-1479, and in part by the Spanish Government under Grant RTI2018-099722-B-I00 (ARISTIDES). The work of K. Zhang and C. Tian was supported in part by the National Science Foundation under Grants CCF-18-32309 and CCF-18-16546.

978-1-5386-6900-6/19/$31.00 ©2019 IEEE

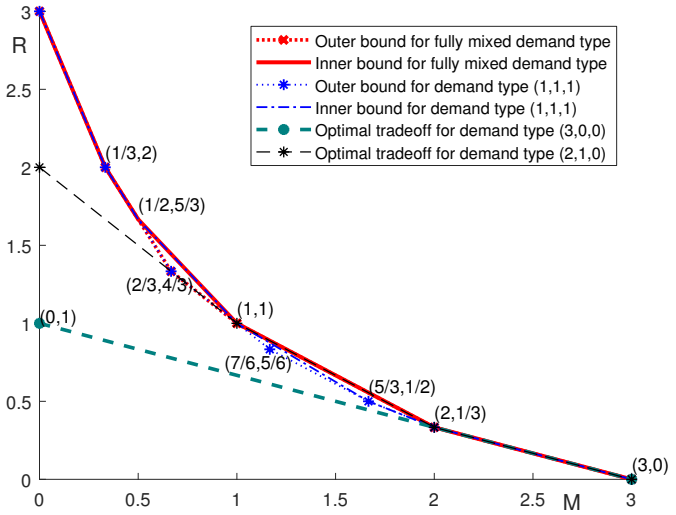

Fig. 1: Inner bounds and outer bounds for $(3,3)$ systems: fully mixed and single demand types.

are requested, and propose a new code construction based on a novel sub-packetization design. The construction is a generalization of the code recently proposed in [8], however, in contrast to the code specific designed for $N = K = 3$ or 4 that yields a single (M, R) pair each, our construction is for general N when $N = K$ which can produce multiple new (M, R) points. Finally, in Section V we consider outer bounds for such single demand type systems, where several existing bounds are first verified to be valid for this relaxed setting, and additionally a new outer bound is identified; the outer bounds indeed match the proposed scheme in some cases.

II. Single Demand Type Systems

In an (N, K) coded caching system, there are N mutually independent uniformly distributed files $(W_1, W_2, \ldots, W_N)$, each of F bits. There are K users, each with a cache memory of capacity MF bits. In the placement phase, each user stores some content, denoted as Z_k for user-k's content, in its cache memory. In the delivery phase, user-k requests a file $d(k)$, and a central server broadcasts a message $X_{d(1),d(2),\ldots,d(K)}$ of rate RF bits to every user, such that each user can decode the requested file, together with the cached contents. The optimal tradeoff between M and R is the fundamental mathematical object of interest.

The notion of demand type was first introduced in [10] (see also [11]), which is restated below.

Definition 1: For a demand vector $(\mathbf{d}(1), \mathbf{d}(2), \ldots, \mathbf{d}(K))$ in an (N, K) coded caching system, denote the number of users requesting file W_n as m_n, where $n \in [1 : N]$. We call the vector obtained by sorting the values $(m_1, m_2, \ldots, m_N)$ in a decreasing order as the demand type of $(\mathbf{d}(1), \mathbf{d}(2), \ldots, \mathbf{d}(K))$.

For example, in an $(N, K) = (3, 3)$ system, the demand vector $(\mathbf{d}(1), \mathbf{d}(2), \mathbf{d}(3)) = (1, 2, 1)$ belongs to the demand type $(2, 1, 0)$. As mentioned earlier, characterizing the fundamental tradeoff between M and R for fully mixed demand type systems appears rather difficult, despite considerable efforts. Hence, we consider a single demand type system in our paper, where it is known a priori that the demand vector must belong to a given type. As a special case, when $N = K$ and the demand type is $(1, 1, \ldots, 1)$, we refer it as a fully demanded coded caching system, or fully demanded system for short.

Our interest is thus to characterize the achievable region of all (M, R) pairs for such single demand type systems, for which the prefetching strategy can be designed accordingly. For example, for the $(N, K) = (3, 3)$ system, the coded caching system for the single demand type $(2, 1, 0)$ only needs to accommodate the following demand vectors

$$(\mathbf{d}(1), \mathbf{d}(2), \mathbf{d}(3)) =$$
$$(1,1,2),(1,2,1),(2,1,1),(2,2,1),(2,1,2),(1,2,2)$$
$$(1,1,3),(1,3,1),(3,1,1),(3,3,1),(3,1,3),(1,3,3)$$
$$(2,2,3),(2,3,2),(3,2,2),(3,3,2),(3,2,3), \text{ or } (2,3,3).$$

III. Fully Mixed and Single Demand Type Systems: The $(3,3)$ Case

In this section, we consider the canonical $(N, K) = (3, 3)$ system, and collect the best known inner bounds and outer bounds for both fully mixed demand type systems and single demand type systems in the literature. This exercise reveals important insight and fundamental differences between the two classes of systems.

A. The Fully Mixed Demand Type System

The best known outer bound for this system can be found in [11], which are all the non-negative pairs of (M, R) satisfying the constraints

$$3M + R \geq 3, 6M + 3R \geq 8, M + R \geq 2,$$
$$2M + 3R \geq 5, M + 3R \geq 3.$$

The lower convex hull of the best known inner bound, on the other hand, is given by the lower convex hull of the points

$$(0, 3), (1/3, 2), (1/2, 5/3), (1, 1), (2, 1/3), (3, 0),$$

where the second and third points are achieved by the scheme in [6], while the others can be achieved by that in [1].

B. Single Demand Type Systems

Next we provide the best known results for the three single demand type systems for $(N, K) = (3, 3)$.

1) For the system with the demand type $(3, 0, 0)$, the achievable region is precisely all the non-negative (M, R) such that
$$M + 3R \geq 3, \tag{1}$$
i.e., in this case, the inner bound and the outer bound match. The outer bound can be obtained by a simple cut-set argument [1], while the inner bound is trivial through a memory-sharing argument.
2) For the system with the demand type $(2, 1, 0)$, the achievable region is precisely all the non-negative (M, R) such that
$$M + R \geq 2, \quad 2M + 3R \geq 5, \quad M + 3R \geq 3.$$

In this case, the corner points $(1,1)$ and $(2,1/3)$ can be achieved using the scheme in [1], and $(0,2)$ and $(3,0)$ are trivial. The outer bound was established in [11].

3) For the system with the demand type $(1,1,1)$, the best known outer bound is given as [11]

$$3M+R\geq 3, 6M+3R\geq 8, M+R\geq 2,$$
$$12M+18R\geq 29, 3M+6R\geq 8, M+3R\geq 3.$$

The lower convex hull of the best known inner bound is given by the lower convex hull of the points

$$(0,3),(1/3,2),(1/2,5/3),(1,1),$$
$$(5/3,1/2),(2,1/3),(3,0).$$

The second and the third points can be achieved by the scheme in [6], the point $(5/3,1/2)$ by the scheme in [8], and the others by that in [1].

C. Fully Mixed vs. Single Demand

By comparing the rate region of different demand type systems in Fig.1, we make the following observations:

1) The point $(5/3,1/2)$, which is achievable for the system with the single demand type $(1,1,1)$ as shown in [8], is in fact not achievable for the $(2,1,0)$ demand type, thus also not achievable for the fully mixed demand system.
2) Between fully mixed and single demand type systems, single demand type systems can indeed achieve lower rates than the fully mixed demand system.
3) Different single demand type systems provide different out bounds for the fully mixed system, with the one with fewer files demanded produce better bounds at high memory regimes, while those with more files being better at low memory regimes.

The first observation implies that the case when all files are requested is not necessarily the "worst case", contrary to popular belief. Thus designing codes for this demand type alone is not sufficient to yield the optimal scheme for the fully mixed demand type systems. Motivated by the observations above and the new code construction in [8], which only provided a single point $N=K=3$ or $N=K=4$, in the sequel we focus on the case $N=K$ for general N, and with a fully demanded system.

IV. Inner Bounds for $(N, K=N)$ Fully Demanded Coded Caching

We propose a novel code construction $(N, K=N)$ for general N values, whose performance is given in the following theorem.

Theorem 1: For an (N,K) caching system with the single demand type $(1,1,\ldots,1)$ and $N=K$, the following rate-memory pairs are achievable.

$$(M,R) \quad = \quad \left(r+\frac{r+1}{K}, \frac{K}{r+1}-1\right), \qquad (2)$$

for $r=\{0,...,K-1\}$.

Note that for $N=3$ and $N=4$, by setting $r=K-2$, we recover the operating points given in [8]; on the other hand, setting $r=0$ gives the point in [7] (see also [4]). Operating points for other values of r are previously unknown to be achievable in this system. In the sequel we provide the code construction, which is also illustrated with an example for $N=K=4$, and $r=2$.

The proposed scheme combines uncoded and coded prefetching. The delivery strategy and the uncoded prefetching part of the scheme follow the original scheme in [1], where coded subfiles are transmitted to simultaneously serve the demands of $r+1$ users. The additional coded prefetching is designed to further exploit the coded transmissions.

A. Prefetching

Let us define the set of all user indexes as $\mathcal{K}=\{1,...,K\}$ and the set of all file indexes as $\mathcal{F}=\{1,...,N\}$. Recall, we have $N=K$. Firstly, we partition each file W_f, for all $f\in\mathcal{F}$ into $K\binom{K-1}{r}$ subfiles of equal size. Each subfile $W_{f,\mathcal{R},s}$ is indexed by an integer s and a set $\mathcal{R}$, with $s\in\mathcal{K}$, $|\mathcal{R}|=r$ and $s\notin\mathcal{R}$, where $r\in\{0,....K-1\}$.

Given a user k, we will prefetch three types of subfiles:

1) Type *I Subfiles:* Place the uncoded subfiles $W_{f,\mathcal{R},s}$ for every file $f\in\mathcal{F}$ and every $\mathcal{R}$ and s satisfying $k\in\mathcal{R}$ and thus $k\neq s$. Observe that there are $m_I=N\binom{K-1}{r-1}(K-r)$ of these subfiles.

2) Type *II Subfiles:* Next, place the coded subfiles

$$Z_{\mathcal{R},k}=\bigoplus_{f\in\mathcal{F}} W_{f,\mathcal{R},k}$$

for all possible set $\mathcal{R}$ satisfying $k\notin\mathcal{R}$. Observe that $m_{II}=\binom{K-1}{r}$ of $Z_{\mathcal{R},k}$ are cached.

3) Type *III Subfiles:* Finally, place the coded subfiles

$$Z_{f,\mathcal{R}^-,k}=\bigoplus_{u\in\mathcal{K}\backslash\{\mathcal{R}^-,k\}} W_{f,u\cup\mathcal{R}^-,k}$$

for each file $f\in\mathcal{F}\backslash g$ and all sets $\mathcal{R}^-$ satisfying $|\mathcal{R}^-|=r-1$ and $\mathcal{R}^-\subset\mathcal{K}\backslash\{k,l\}$, where g and l are any arbitrary user and file indexes, respectively. It can be shown, that user k can obtain $Z_{f,\mathcal{R}^-,k}$ for all $f\in\mathcal{F}$ and all $\mathcal{R}^-\subset\mathcal{K}\backslash\{k\}$, from the subfiles cached. Thus, the total number of these coded subfiles is $m_{III}=(K-1)\binom{K-2}{r-1}$.

The scheme caches m_I uncoded subfiles, m_{II} coded subfiles $Z_{\mathcal{R},s}$ and m_{III} coded subfiles $Z_{f,\mathcal{R}^-,s}$. Because each subfile has $\frac{F}{K\binom{K-1}{r}}$ bits, the required cache load at users equals MF with $M=\frac{m_I+m_{II}+m_{III}}{K\binom{K-1}{r}}=r+\frac{r+1}{K}$.

B. Delivery

Consider the delivery transmission for a fixed s. For every set $\mathcal{R}^+\subset\mathcal{K}\backslash s$ with $r+1$ users, i.e. $|\mathcal{R}^+|=r+1$, the server broadcasts

$$Y_{\mathcal{R}^+,s} \quad = \quad \bigoplus_{u\in\mathcal{R}^+} W_{\mathbf{d}(u),\mathcal{R}^+\backslash u,s}.$$

Since $s\notin\mathcal{R}^+$, the total number of transmissions associated to a given s is $\binom{K-1}{r+1}$, and the total number of transmission is

$$T \quad = \quad \sum_{s\in\mathcal{K}}\binom{K-1}{r+1}=K\binom{K-1}{r+1},$$

TABLE I: Prefetched Symbols at User 1 for Caching System $(N = 4, K = 4), r = 2$

	Cached Symbols at User 1
Type I	$W_{1,12,3}, W_{1,12,4}, W_{1,13,2}, W_{1,13,4}, W_{1,14,2}, W_{1,14,3}$ $W_{2,12,3}, W_{2,12,4}, W_{2,13,2}, W_{2,13,4}, W_{2,14,2}, W_{2,14,3}$ $W_{3,12,3}, W_{3,12,4}, W_{3,13,2}, W_{3,13,4}, W_{3,14,2}, W_{3,14,3}$ $W_{4,12,3}, W_{4,12,4}, W_{4,13,2}, W_{4,13,4}, W_{4,14,2}, W_{4,14,3}$
Type II	(1) $W_{1,23,1} \oplus W_{2,23,1} \oplus W_{3,23,1} \oplus W_{4,23,1}$ (2) $W_{1,24,1} \oplus W_{2,24,1} \oplus W_{3,24,1} \oplus W_{4,24,1}$ (3) $W_{1,34,1} \oplus W_{2,34,1} \oplus W_{3,34,1} \oplus W_{4,34,1}$
Type III	(4) $W_{2,23,1} \oplus W_{2,34,1}$, (5) $W_{3,23,1} \oplus W_{3,34,1}$ (6) $W_{4,23,1} \oplus W_{4,34,1}$, (7) $W_{2,24,1} \oplus W_{2,34,1}$ (8) $W_{3,24,1} \oplus W_{3,34,1}$, (9) $W_{4,24,1} \oplus W_{4,34,1}$

TABLE II: Delivery for Demand $\mathbf{d} = (W_1, W_2, W_3, W_4)$

Delivery
(10) $W_{2,34,1} \oplus W_{3,24,1} \oplus W_{4,23,1}$
(11) $W_{1,34,2} \oplus W_{3,14,2} \oplus W_{4,13,2}$
(12) $W_{1,24,3} \oplus W_{2,14,3} \oplus W_{4,12,3}$
(13) $W_{1,23,4} \oplus W_{2,13,4} \oplus W_{3,12,4}$

and over the number of each file's subfiles $\frac{F}{K\binom{K-1}{r}}$, the communication load RF is

$$R = \frac{\binom{K}{1}\binom{K-1}{r+1}}{K\binom{K-1}{r}} = \frac{K}{r+1} - 1. \quad (3)$$

C. Decoding Subfiles Uncoded at Users Different from u

Firstly we consider the decoding at user u of subifiles $W_{\mathbf{d}(u),\mathcal{R},s}$ with $u \neq s$. If $u \in \mathcal{R}$, then file $W_{\mathbf{d}(u),\mathcal{R},s}$ can be found uncoded at the cache of user u. Instead, if $u \notin \mathcal{R}$ then user u computes

$$\begin{aligned} &W_{\mathbf{d}(u),\mathcal{R},s} \\ =&W_{\mathbf{d}(u),\mathcal{R},s} \oplus \bigoplus_{j\in\mathcal{R}} W_{\mathbf{d}(j),\{\mathcal{R}\cup u\}\backslash j,s} \oplus \bigoplus_{i\in\mathcal{R}} W_{\mathbf{d}(i),\{\mathcal{R}\cup u\}\backslash i,s} \\ =&Y_{\mathcal{R}\cup u,s} \oplus \bigoplus_{i\in\mathcal{R}} W_{\mathbf{d}(i),\{\mathcal{R}\cup u\}\backslash i,s} \end{aligned}$$

using $Y_{\mathcal{R}\cup u,s}$ and the subfiles $W_{\mathbf{d}(i),\{\mathcal{R}\backslash i\}\cup u,s}$ for all $i \in \mathcal{R}$ that are uncoded in the cache of user u.

D. Decoding Subfiles Coded at the Cache of User u

Next, we show how user u obtains $W_{\mathbf{d}(u),\mathcal{R},u}$ for all $\mathcal{R}$. Recall that these subfiles are all coded in coded subfiles of its own cache. First user s computes

$$\begin{aligned} \bigoplus_{v\notin\mathcal{R}\cup u} Y_{\mathcal{R}\cup v,u} &= \bigoplus_{v\notin\mathcal{R}\cup u} \bigoplus_{t\in\mathcal{R}\cup v} W_{\mathbf{d}(t),\{\mathcal{R}\cup v\}\backslash t,u} \\ &= \bigoplus_{v\notin\mathcal{R}\cup u} W_{\mathbf{d}(v),\mathcal{R},s} \\ &\quad \oplus \bigoplus_{t\in\mathcal{R}} W_{\mathbf{d}(t),\mathcal{R},s} \bigoplus_{v\notin\mathcal{R}\backslash t} W_{\mathbf{d}(t),\{\mathcal{R}\backslash t\}\cup v,s}. \end{aligned}$$

Recall that user u caches

$$Z_{\mathbf{d}(t),\mathcal{R}\backslash t} = \bigoplus_{v\notin\mathcal{R}\backslash t} W_{\mathbf{d}(t),\{\mathcal{R}\backslash t\}\cup v,u}.$$

Thus, user u can compute

$$\begin{aligned} \Omega_{\mathcal{R},u} &= \bigoplus_{v\notin\mathcal{R}\cup u} Y_{\mathcal{R}\cup v} \oplus \bigoplus_{t\in\mathcal{R}} Z_{\mathbf{d}(t),\mathcal{R}\backslash t} \\ &= \bigoplus_{v\notin\mathcal{R}\cup u} W_{\mathbf{d}(v),\mathcal{R},u} \oplus \bigoplus_{t\in\mathcal{R}} W_{\mathbf{d}(t),\mathcal{R},u} \\ &= W_{\mathbf{d}(u),\mathcal{R},u} \oplus \bigoplus_{f\in\mathcal{F}} W_{f,\mathcal{R},u}, \end{aligned}$$

where $\bigoplus_{f\in\mathcal{F}} W_{f,\mathcal{R},u}$ is cached in user u as well, hence $W_{\mathbf{d}(u),\mathcal{R},u}$ can be decoded by the user u.

E. An Example: $(N, K) = (4, 4)$ and $r = 2$

Set $r = 2$, and thus there are $\binom{K}{r}\binom{K-r}{1} = 12$ subfiles per file. This corresponds to the point $(M, R) = (11/4, 1/3)$, which can be achieved by the scheme in [8]. We now present a new approach using the proposed scheme. The prefetching strategy is shown in Table I, which contains all cached symbols in User 1. Consider demand (W_1, W_2, W_3, W_4), our delivery strategy yields a transmission of four coded symbols, as specified in Table II. The decoding process is as follows. First, observe that, the 6 requested subfiles $W_{1,13,2}, W_{1,14,2}, W_{1,12,3}, W_{1,14,3}, W_{1,12,4}, W_{1,13,4}$ are cached uncoded by User 1. Second, the 3 subfiles $W_{1,34,2}, W_{1,34,2}, W_{1,34,2}$ can be directly decoded from the three transmissions (11-13), respectively, since in each transmission all other subfiles are stored uncoded by User 1. Third, we can use the transmission (10), to recover the remaining 3 subfiles of W_1, which are coded at the cache of User 1, as

$$\begin{aligned} W_{1,23,1} &= (1) \oplus (4) \oplus (5) \oplus (8) \oplus (10), \\ W_{1,24,1} &= (2) \oplus (6) \oplus (7) \oplus (9) \oplus (10), \\ W_{1,34,1} &= (3) \oplus (6) \oplus (8) \oplus (10). \end{aligned}$$

Hence all the 12 subfiles of W_1 are successfully recovered by User 1. The decoding steps for the other users follows the same pattern.

V. General Outer Bounds and Evaluation

In this section, we consider the outer bounds of the fully demanded system. We first verify the validity of several existing outer bounds derived for the fully mixed demand type system, in the context of this single demand type system setting. Then, a new outer bound is provided, which is only applicable to the fully demanded system.

A. General Outer Bounds for Fully Demanded System

In this subsection we give out two outer bounds in Theorem 2 and Theorem 3. Theorem 2 is induced from the outer bounds of the original fully mixed demand type system in [12], and Theorem 3 is identified merely for the fully demanded coded caching system.

Two different sets of outer bounds were given in [12]. One is obtained by the intersection of outer bounds derived using single demand types, hence only one particular outer bound remains valid for our setting which however becomes trivial, another set of outer bounds still holds, as its core inequality

$$RF \geq \sum_{k=1}^{K} H(W_k | Z_{[1:k]}, W_{[1:k-1]})$$

holds exactly for the fully demanded coded caching system.

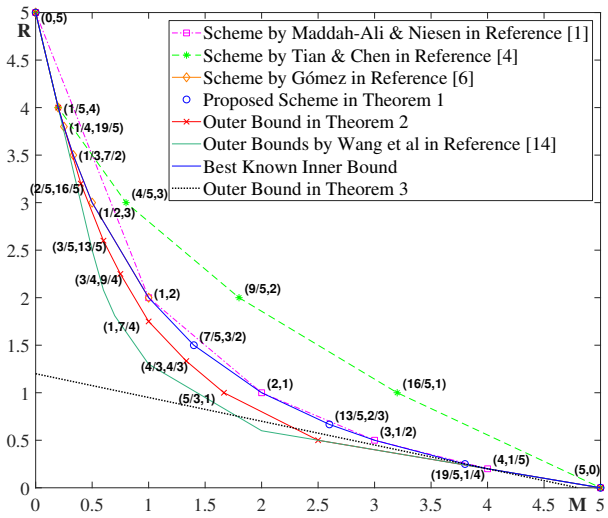

Fig. 2: Inner bounds and outer bounds for the $(5,5)$ system with demand type $(1,1,1,1,1)$.

Therefore, we have the following theorem.

Theorem 2 (Yu et al. [12]): In the $(N, K = N)$ coded caching system with demand type $(1,1,\ldots,1)$, all achievable (M,R) rate pairs are lower-bounded by the lower convex envelop of the points

$$\left(\frac{K-\ell+1}{s}, \frac{s-1}{2}+\frac{\ell(\ell-1)}{2s}\right),$$

where $s \in \{1,2,\ldots,K\}$ and $\ell \in 0,1,\ldots,K$.

Similarly, the outer bound of the worst case rate in [14] is also valid for this setting. However it is weaker than Theorem 2 (see Fig. 2), hence the details are omitted here. We can also prove the following outer bound.

Theorem 3: In the $(N, K = N)$ coded caching system with demand type $(1,1,\ldots,1)$, all achievable (M,R) rate pairs must satisfy:

$$KM + K(K-1)R \geq K^2 - 1. \tag{4}$$

The proof of this bound is omitted due to space constraint. We note that [8], a similar outer bound was established for $N = 3,4$, and Theorem 3 generalizes those bounds. When setting $r = K-2$ in Theorem 1, the achievable rate pair $(M,R) = (K-2+\frac{K-1}{K}, \frac{1}{K-1})$ indeed matches this outer bound, hence optimal for the fully demanded system.

B. Example Evaluation: $(N,K) = (5,5)$

When $(N,K) = (5,5)$, Theorem 2 reduces to

$$\begin{array}{lll} 5M+R \geq 5, & 20M+5R \geq 24, & 36M+10R \geq 47, \\ 6M+2R \geq 9, & 2M+R \geq 4, & 5M+4R \geq 13, \\ 5M+6R \geq 16, & 5M+9R \geq 19, & 5M+12R \geq 21, \\ 5M+16R \geq 23, & 5M+20R \geq 24, & M+5R \geq 5. \end{array}$$

and Theorem 3 reduces to

$$5M+20R \geq 24.$$

The inner bounds and outer bounds are shown in Fig. 2. It can be seen that three new operating points $(7/5, 3/2)$, $(13/5, 2/3)$ and $(19/5, 1/4)$ are obtained by the proposed code construction.

VI. Conclusion

We considered the single demand type coded caching systems in this work. For the canonical $(3,3)$ system, the single demand type systems are compared thoroughly with fully mixed demand type systems. Even in this case, we see that codes designed for the demand type $(1,1,1)$ in fact operates strictly outside of the achievable region of the $(2,1,0)$. This is contrary to the popular belief that the fully demanded system is "the worst case". We then proposed a new scheme for the fully demanded system, which can achieve the rate pair $(M,R) = (r+\frac{r+1}{K}, \frac{K}{1+r}-1)$ with $r \in [0:K-1]$. Lastly, we adapted several outer bounds original obtained for the fully mixed demand type systems, and also identified a new outer bound specific for the single demand type (fully demanded) system. The proposed code construction provide operating point that indeed match the outer bounds in some cases.

Acknowledgment

The authors wish to thank K.P. Vijith Kumar for sharing a preprint of the paper [8].

References

[1] M. A. Maddah-Ali and U. Niesen, "Fundamental Limits of Caching," *IEEE Trans. on Information Theory*, vol. 60, no. 5, pp. 2856-2867, May 2014.

[2] Q. Yu, M. A. Maddah-Ali, and A. S. Avestimehr, "The Exact Rate-Memory Tradeoff for Caching With Uncoded Prefetching," *IEEE Trans. on Information Theory*, vol. 64, no. 2, pp. 1281-1296, Feb. 2018.

[3] S. Sahraei and M. Gastpar, "K Users Caching Two Files: An Improved Achievable Rate," in *Proc. 2016 Annual Conference on Information Science and Systems (CISS)*, Princeton, NJ, 2016, pp. 620-624.

[4] C. Tian and J. Chen, "Caching and Delivery via Interference Elimination," *IEEE Trans. on Information Theory*, vol. 64, no. 3, pp. 1548-1560, March 2018.

[5] K. Zhang and C. Tian. "Fundamental Limits of Coded Caching: From Uncoded Prefetching to Coded Prefetching." *IEEE Journal on Selected Areas in Communications*, vol. 36, no. 6, pp. 1153-1164, Jun. 2018.

[6] J. Gómez-Vilardebó, "Fundamental Limits of Caching: Improved Rate-Memory Tradeoff With Coded Prefetching," *IEEE Trans. on Communications*, vol. 66, no. 10, pp. 4488-4497, Oct. 2018.

[7] Z. Chen, P. Fan, and K. B. Letaief, "Fundamental Limits of Caching: Improved Bounds for Users with Small Buffers," *IET Communications*, vol. 10, no. 17, pp. 2315-2318, Nov. 2016.

[8] K.P. Vijith Kumar, B. Kumar, and T. Jacob, "Towards the Exact Rate Memory Tradeoff in Coded Caching," in *Proc. National Conference on Communication (NCC) 2019*, Bangalore.

[9] M. Mohammadi Amiri and D. Gündüz, "Fundamental Limits of Coded Caching: Improved Delivery Rate-Cache Capacity Tradeoff," *IEEE Trans. on Communications*, vol. 65, no. 2, pp. 806-815, Feb. 2017.

[10] C. Tian, "Symmetry, Demand Types and Outer Bounds in Caching Systems," in *Proc. 2016 IEEE International Symposium on Information Theory (ISIT)*, Barcelona, 2016, pp. 825-829.

[11] C. Tian, "Symmetry, Outer Bounds, and Code Constructions: A Computer-Aided Investigation on the Fundamental Limits of Caching," *MDPI Entropy*. 2018; 20(8):603.

[12] Q. Yu, M. A. Maddah-Ali, and A. S. Avestimehr, "Characterizing the Rate-Memory Tradeoff in Cache Networks Within a Factor of 2," *IEEE Trans. on Information Theory*, vol. 65, no. 1, pp. 647-663, Jan. 2019.

[13] H. Ghasemi and A. Ramamoorthy, "Improved Lower Bounds for Coded Caching," *IEEE Trans. on Information Theory*, vol. 63, no. 7, pp. 4388-4413, July 2017.

[14] C. Wang, S. Saeedi Bidokhti, and M. Wigger, "Improved Converses and Gap Results for Coded Caching," *IEEE Trans. on Information Theory*, vol. 64, no. 11, pp. 7051-7062, Nov. 2018.