Coded Caching with Full Heterogeneity: Exact Capacity of The Two-User/Two-File Case

Chih-Hua Chang and Chih-Chun Wang, Purdue ECE, USA

***Abstract*—The most commonly used setting in the coded caching literature consists of the following five elements: (i) homogeneous file sizes, (ii) homogeneous cache sizes, (iii) user-independent homogeneous file popularity (i.e., all users share the same file preference), and (iv) worst-case rate analysis. While recent results have relaxed some of these assumptions, deeper understanding of the full heterogeneity setting is still much needed since traditional caching schemes place little assumptions on file/cache sizes and almost always allow each user to have his/her own file preference through individualized file request prediction. Taking a microscopic approach, this paper characterizes the *exact capacity* of the smallest 2-user/2-file ($N = K = 2$) problem but under the most general setting that simultaneously allows for (i) heterogeneous files sizes, (ii) heterogeneous cache size, (iii) user-dependent file popularity, and (iv) average-rate analysis. Solving completely the case of $N = K = 2$, the results would shed further insights on the performance and complexity of optimal coded caching with full heterogeneity for arbitrary N and K.**

I. Introduction

Recently [1] shows that coded caching can shorten the *worst-case* delivery time by a factor of $(\frac{1}{1+KM/FN})$ when compared to the traditional uncoded caching schemes, where N is the number of files, K is the number of users, M is the individual cache size and F is the individual file size. Although the capacity of the general coded caching problem remains an open problem, the optimal coded caching scheme (exact capacity) is characterized for some special cases [1]–[3] and order-optimal capacity characterization for several more general scenarios [1], [4]–[8].

Most existing results are based on the settings of (i) homogeneous file sizes, (ii) homogeneous cache size, (iii) user-independent homogeneous file popularity, and (iv) worst-case analysis. These settings are not 100% compatible with the traditional uncoded caching solutions. Specifically, the basic design principle of traditional schemes is to first predict the likelihood of the next file request for each individual user (i.e., user-dependent heterogeneous file popularity), then let each user store the most likely file(s) until his/her cache is full (which is naturally applicable to heterogeneous file and cache sizes). The rationale behind is that such probability-based greedy solution would reduce the *average rate* during delivery, even though there is no optimality guarantee.

Because of the aforementioned differences between their settings, a coded scheme designed for the homogeneous, worst-case setting could have significantly worse average-rate performance in practice when compared to a traditional scheme, especially for the scenarios in which the individualized file request prediction is very effective and the file and cache sizes are highly heterogeneous.

In principle, since coded caching is a strict generalization of any uncoded solution, an optimal coded caching solution should outperform its non-coded counterpart under *any setting*. This potential loss of performance is mainly due to the homogeneous settings on which existing coded caching schemes are optimized, which mismatch the practical scenarios.

Motivated by this observation, this work studies the exact capacity region and the corresponding optimal coded caching schemes under (i) heterogeneous file sizes, (ii) heterogeneous cache size, (iii) user-dependent heterogeneous file popularity, and (iv) average-case analysis. Such results, if successful, would allow the system designers to accurately assess the performance gain of coded caching (the ultimate capacity minus the achievable rate of traditional uncoded schemes) in a practical heterogeneous setting. While the problem remains open for general N and K values, we characterize the exact capacity for $N = K = 2$. The results would shed further insight for general N and K.

A. Comparison to Existing Results

Several existing works relaxed parts of the above conditions (i) to (iv). Table I provides a non-comprehensive list of several related results. As can be seen in Table I, finding the exact capacity remains a difficult task and most existing exact capacity results are based on small K (i.e., $K = 2$ or $K = 3$) and focus on the *worst-case* rate rather than a general probabilistic average rate model, e.g., [1]–[3], [9]. One of the most general heterogeneous setting results is [10], which uses linear programming results to search for better achievable rates without deriving any converse bounds, and is not focused on the general user-dependent file popularity setting. To the best of our knowledge, there exists neither exact capacity nor order-optimal results for the average rate setting with heterogeneous file/cache sizes and user-dependent file popularity.

II. Problem Formulation

We consider the simplest non-trivial coded caching system with $N = 2$ files and $K = 2$ users. A central server has access to two files W_1 and W_2 of file sizes F_1 and F_2 bits, respectively. The cache content of user k is denoted by Z_k and is of size M_k for $k \in \{1, 2\}$. Without loss of generality, we assume $M_k \in [0, F_1 + F_2]$ for all k.

In the *placement phase*, user k populates its cache by

$$Z_k = \phi_k(W_1, W_2), \tag{1}$$

where ϕ_k is the caching function of user k. In the *delivery phase*, the two users send a demand request $(d_1, d_2) \in \{1, 2\}^2$

This work was supported in parts by NSF under Grant CCF-1422997, Grant ECCS-1407604, Grant CCF-1618475, and Grant CCF-1816013.

TABLE I
COMPARISONS OF EXISTING RESULTS

	Worst-case rate	Average rate
Homo. file and homo. cache sizes	Arbitrary K and N, order-optimal rate [1], [4], [8] $K=2$ and arbitrary N, exact capacity [1], [2] $K=3$ and $N=2$, exact capacity [2]	Arbitrary K and N, order-optimal rate [5]–[8] Arbitrary K and N, achievable rate only [11], [12]
Homo. file and heter. cache sizes	$K=2$ and arbitrary N, exact capacity [3]	Arbitrary K and N, achievable rate only [10]
Heter. file and homo. cache sizes	Arbitrary K and N, order-optimal rate [13]	Arbitrary K and N, achievable rate only [10]
Heter. file and heter. cache sizes	$N=K=2$, exact capacity [9]	Arbitrary K and N, achievable rate only [10]

to the server, i.e., user k demands file W_{d_k}. The probability mass function of the demand request $\vec{d} \triangleq (d_1, d_2)$ is denoted by $p_{\vec{d}}$, which satisfies $\sum_{\vec{d}\in\{1,2\}^2} p_{\vec{d}} = 1$. We assume $\{p_{\vec{d}}\}$, which can be of any arbitrary value, is known to the server.

After receiving $\vec{d}$, the server *broadcasts* an encoded signal

$$X_{\vec{d}} = \psi(\vec{d}, W_1, W_2) \tag{2}$$

of $R_{\vec{d}}$ bits with encoding function ψ. Each user k then uses its cache content Z_k to decode his/her desired file

$$\hat{W}_{d_k} = \mu_k(\vec{d}, X_{\vec{d}}, Z_k) \tag{3}$$

where μ_k is the decoding function of user k.

Definition 1. *A coded caching scheme is specified completely by its five functions $\{\phi_1, \phi_2, \psi, \mu_1, \mu_2\}$ and it is* zero-error feasible *if $\hat{W}_{d_k} = W_{d_k}$ for all $\vec{d} \in \{1,2\}^2$, all $k \in \{1,2\}$, and all $W_k \in \{0,1\}^{F_k}$.*

Definition 2. *The worst-case rate of a zero-error coded caching scheme is*

$$R^* = \max_{\vec{d}\in\{0,1\}^2} R_{\vec{d}}. \tag{4}$$

The worst-case capacity is the infimimum of the worst-case rates of all zero-error schemes.

Definition 3. *The average rate of a zero-error coded caching scheme is*

$$\bar{R} = \sum_{\vec{d}\in\{1,2\}^2} p_{\vec{d}} R_{\vec{d}}. \tag{5}$$

The average-rate capacity is the infimimum of the average rates of all zero-error schemes.

For simplicity, we slightly abuse the above notation and directly use R^* and $\bar{R}$ to denote the worst-case and the average-rate capacities, respectively, even though their original notations in (4) and (5) are for the achievable rates instead.

III. MAIN RESULTS

Define the following strictly more general concept.

Definition 4. *The* per-request capacity region *(PRCR) is the closure of the rate vectors $\vec{R} = (R_{(1,1)}, R_{(1,2)}, R_{(2,1)}, R_{(2,2)})$ of all zero-error coded caching schemes.*

PRCR is the most fundamental performance limits of coded caching since it captures the optimal tradeoff needed to simultaneously satisfy different request patterns. Our main result is an exact characterization of the PRCR. In the end of Section III-B, we also elaborate how to use the PRCR to solve R^* and $\bar{R}$.

A. Lower Bounds of the PRCR

We derive the following lower bounds for arbitrary file and cache sizes (F_1, F_2, M_1, M_2).

Instance 0: Nonnegative rates:

$$R_{\vec{d}} \geq 0, \quad \forall \vec{d} \in \{1,2\}^2. \tag{6}$$

By varying $\vec{d}$, there are totally 4 inequalities in Instance 0.

Instance 1: For any $i, j \in \{1,2\}$, there are two inequalities:

$$R_{(i,j)} + M_1 \geq F_i, \text{ and } R_{(i,j)} + M_2 \geq F_j. \tag{7}$$

By varying i, j, there are totally 8 inequalities in Instance 1.

Instance 2: For any $(i,j) = (1,2)$ or $(2,1)$,

$$R_{(i,j)} + M_1 + M_2 \geq F_1 + F_2. \tag{8}$$

By varying (i,j), there are totally 2 inequalities in Instance 2.

Instance 3: For any $i, j \in \{1,2\}$, there are two inequalities:

$$R_{(i,1)} + R_{(j,2)} + M_2 \geq F_1 + F_2, \tag{9}$$

$$R_{(1,i)} + R_{(2,j)} + M_1 \geq F_1 + F_2. \tag{10}$$

By varying i, j, there are totally 8 inequalities in Instance 3.

Instance 4 uses a more refined technique[1] and we thus provide the detailed derivation.

Instance 4: For any $(i,j) = (1,2), (2,1)$, or $(2,2)$,

$$\begin{aligned}
&R_{(i,1)} + R_{(1,j)} + M_1 + M_2 && (11)\\
&\geq H(X_{(i,1)}) + H(Z_2) + H(X_{(1,j)}) + H(Z_1) && (12)\\
&\geq H(X_{(i,1)}, Z_2) + H(X_{(1,j)}, Z_1) && (13)\\
&\geq H(X_{(i,1)}, Z_2, W_1) + H(X_{(1,j)}, Z_1, W_1) && (14)\\
&\geq H(X_{(i,1)}, X_{(1,j)}, Z_1, Z_2, W_1) + H(W_1) && (15)\\
&\geq H(X_{(i,1)}, X_{(1,j)}, Z_1, Z_2, W_1, W_2) + H(W_1) && (16)\\
&= H(W_1, W_2) + H(W_1) = 2F_1 + F_2 && (17)
\end{aligned}$$

where (13) follows from that the sum of marginal entropies is no less than the joint entropy; (14) follows from that user 2 can decode W_1 based on $X_{(i,1)}$ and Z_2 and user 1 can decode W_1 based on $X_{(1,j)}$ and Z_1; (15) follows from the Shannon-type inequality; (16) follows from that we can decode W_2 from $X_{(i,1)}$, $X_{(1,j)}$, Z_1, and Z_2; and (17) follows from that X's and Z's are functions of (W_1, W_2).

Symmetrically for any $(i,j) = (1,2), (2,1)$, or $(1,1)$

$$R_{(i,2)} + R_{(2,j)} = M_1 + M_2 \geq F_1 + 2F_2.$$

Varying (i,j), there are totally 6 inequalities in Instance 4.

Totally, there are 28 linear inequalities in Instances 0 to 4.

[1]A more general version of the techniques can be found in [1], [2], [14].

B. Coded Caching Capacity for $N = K = 2$

The derivation of the aforementioned lower bounds is relatively straightforward, see [1], [2], [9], [14]. A more important contribution of this work is to show that these lower bounds indeed characterize the exact 4-dimensional PRCR.

Proposition 1. *Consider arbitrary (F_1, F_2, M_1, M_2). For any $\vec{R}$ that satisfies the 28 lower bounds in Section III-A simultaneously, we can find a zero-error scheme attaining such $\vec{R}$.*

The proof of Proposition 1 is based on the following lemma.

Lemma 1. *The 4-D polytope formed by the 28 linear inequalities has either 2 or 4 or 6 distinct corner points. The actual number depends on the underlying (F_1, F_2, M_1, M_2) value. An exhaustive list of all the corner points is provided jointly in Fig. 1 and Table II.*

Proposition 2. *All 28 corner points listed in Fig. 1 and Table II can be achieved with explicit code construction.*

Lemma 1 and Proposition 2 jointly imply Proposition 1.

Due to space limits, in Section III-C we prove Lemma 1 for the sub-case "$0 \le M_2 \le M_1$ and $M_1 + M_2 \le F_2$", which has exactly 4 corner points Vertices 1, 2, 3, and 4, see Fig 1. Among them, the most interesting achievable scheme is for Vertex 4 and we thus also provide its details in Section III-C. A complete and detailed proof for all the cases and all the achievable schemes can be found in [15].

Knowing all corner points of the per-request capacity region (PRCR) is very beneficial. Since the average capacity corresponds to the vector $\vec{R}$ in the PRCR that has the smallest linear objective value $\sum p_{\vec{d}} R_{\vec{d}}$ and since the minimum of a linear programming problem can only happen at the corner points, we can easily use the corner points in Fig. 1 and Table II to characterize the average-rate capacity of arbitrary popularity vector $(p_{(1,1)}, p_{(1,2)}, p_{(2,1)}, p_{(2,2)})$. Namely, given any (F_1, F_2, M_1, M_2), we first use Fig. 1 to figure out all the corner points in the RPCR. Then for each corner point, we plug in the closed-form expression in Table II to the objective function $\sum p_{\vec{d}} R_{\vec{d}}$. Repeat this process for each corner point. Finally the smallest objective function must be the average-rate capacity under the given (F_1, F_2, M_1, M_2) and $(p_{(1,1)}, p_{(1,2)}, p_{(2,1)}, p_{(2,2)})$. Two example results of this general procedure are provided as follows.

Corollary 1. *For arbitrary (F_1, F_2) and uniform file popularity (i.e., $p_{\vec{d}} = 0.25$, $\forall \vec{d}$), the average-rate capacity for arbitrary (M_1, M_2) is described in Fig. 2.*

Corollary 2. *Suppose $(F_1, F_2) = (1.5, 1)$ and user 1 demands files 1 and 2 with probability 2/3 and 1/3, respectively, and user 2 demands files 1 and 2 with probability 0.4 and 0.6, respectively, and the demands of the users are independent. The corresponding average-rate capacity for arbitrary (M_1, M_2) is described in Fig. 3.*

The exact PRCR characterization can also be used to easily rederive the worst-cast capacity R^* with arbitrary (F_1, F_2, M_1, M_2), previously found by examining the outer bounds of entropic cones [9]. See [15] for details.

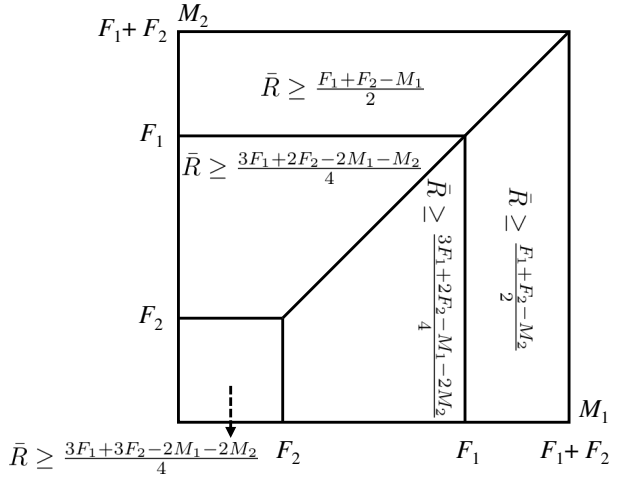


Fig. 2. The average-rate capacity of uniform popularity.

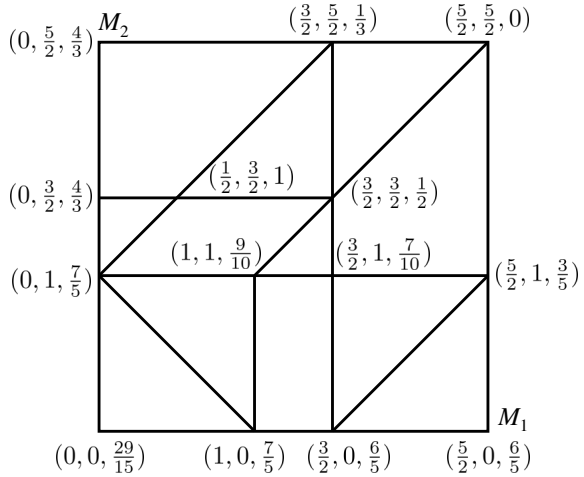


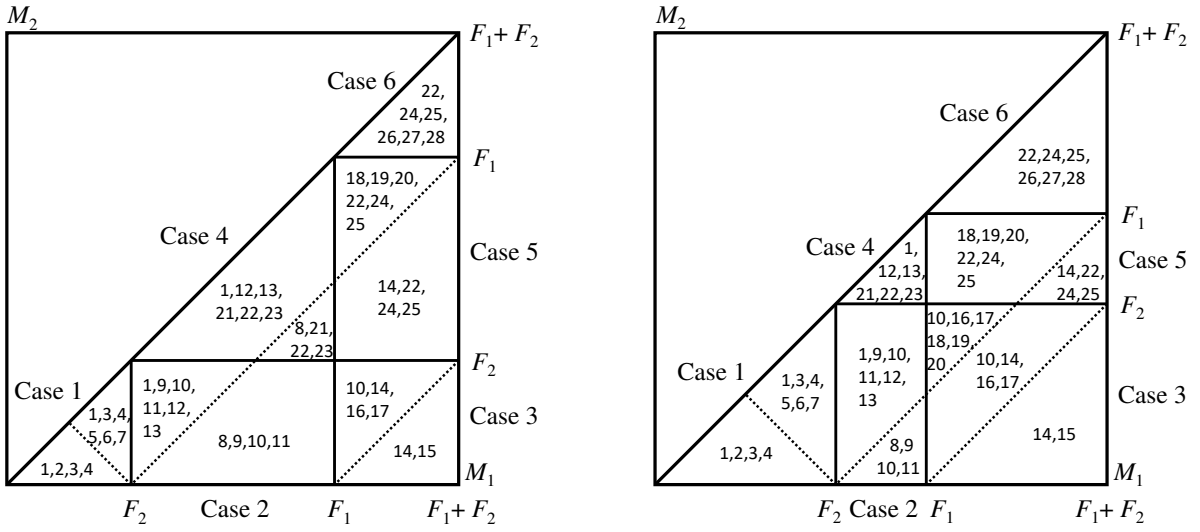
Fig. 3. The average-rate capacity with $(F_1, F_2) = (1.5, 1)$ and $(p_{(1,1)}, p_{(1,2)}, p_{(2,1)}, p_{(2,2)}) = (\frac{4}{15}, \frac{2}{5}, \frac{2}{15}, \frac{1}{5})$. There are 12 facets and 14 corner points. Each corner point is labeled by a tuple $(M_1, M_2, \bar{R})$, where (M_1, M_2) describe the location and the third coordinate describe the corresponding exact average-rate capacity $\bar{R}$. The capacity is asymmetric with respect to (M_1, M_2) due to the heterogeneous file popularity.

The closed form expressions of R^* and $\bar{R}$ as functions of (F_1, F_2, M_1, M_2) and $\{p_{\vec{d}}\}$, e.g., Corollaries 1 and 2, can be used to solve other design optimization problems. For example, we can solve the 2-user/2-file *memory allocation problem* [16] optimally by finding the (M_1^*, M_2^*) that minimizes R^* (or $\bar{R}$) subject to the total memory constraint $M_1 + M_2 \le M_{\text{total}}$.

C. Sketch Of The Proofs

We now sketch the proofs for the case "$0 \le M_2 \le M_1$ and $M_1 + M_2 \le F_2$". To that end, we first summarize and simplify the 14 inequalities in Instances 0 to 2 as

$$R_{(1,1)} \ge F_1 - M_2 \tag{A1}$$
$$R_{(1,2)} \ge F_1 + F_2 - M_1 - M_2 \tag{A2}$$
$$R_{(2,1)} \ge F_1 + F_2 - M_1 - M_2 \tag{A3}$$
$$R_{(2,2)} \ge F_2 - M_2 \tag{A4}$$



(a) $F_1 \geq 2F_2$ (b) $F_1 < 2F_2$

Fig. 1. Description of the regions of (M_1, M_2) and the corresponding corner points. The x-axis (resp. y-axis) is for the M_1 (resp. M_2) value. In this figure we assume $F_1 \geq F_2$ and only describe the cases when $M_1 \geq M_2$, thus the lower-half of the line $M_1 = M_2$. The cases of $F_1 < F_2$ and $M_2 < M_1$ can be obtained by swapping the file and user indices, respectively. Two scenarios are considered: (a) If $F_1 \geq 2F_2$; (b) If $F_1 < 2F_2$. In both scenarios, there are 6 major regions described by solid lines, which are labeled as Cases 1 to 6. The 6 major regions are further partitioned into 11 sub-regions by three 45-degree dotted lines. The numbers within each sub-regions are the indices of the corner points of the 4-D PRCR polytope when (M_1, M_2) falls into the sub-region. For example, in both (a) and (b), the triangular subregion corresponding to "$0 \leq M_2 \leq M_1$ and $M_1 + M_2 \leq F_2$" are labeled by "1,2,3,4". This means that when "$0 \leq M_2 \leq M_1$ and $M_1 + M_2 \leq F_2$" holds, the 4 corner points are vertices 1, 2, 3, and 4 described in Table II.

TABLE II
THE EXPRESSIONS OF ALL POSSIBLE CORNER POINTS. WE USE $F_{1+2} \triangleq F_1 + F_2$ AND $M_{1+2} \triangleq M_1 + M_2$ AS SHORTHAND

	Closed-form expression: $\vec{R} = (R_{(1,1)}, R_{(1,2)}, R_{(2,1)}, R_{(2,2)})$		Closed-form expression: $\vec{R} = (R_{(1,1)}, R_{(1,2)}, R_{(2,1)}, R_{(2,2)})$
1	$(F_1 - M_2, F_1 + F_2 - M_1, F_1 + F_2 - M_1, F_2)$	15	$(F_1, F_2 - M_2, F_1, F_2 - M_2)$
2	$(F_1, F_1 + F_2 - M_1 - M_2, F_1 + F_2 - M_1 - M_2, F_2)$	16	$(F_1, F_2 - M_2, F_1, F_1 + F_2 - M_1)$
3	$(F_1, F_1 + F_2 - M_1, F_1 + F_2 - M_1, F_2 - M_2)$	17	$(\frac{F_1+M_1-M_2}{2}, F_2 + \frac{F_1-M_{1+2}}{2}, \frac{F_1+M_1-M_2}{2}, F_2 + \frac{F_1-M_{1+2}}{2})$
4	$(F_1 - \frac{M_2}{2}, F_{1+2} - M_1 - \frac{M_2}{2}, F_{1+2} - M_1 - \frac{M_2}{2}, F_2 - \frac{M_2}{2})$	18	$(F_1 - M_2, F_2, F_1 + F_2 - M_1, F_2)$
5	$(F_1, F_1 + F_2 - M_1 - M_2, F_1, F_2)$	19	$(F_1 + F_2 - M_1, F_2, F_1 - M_2, F_2)$
6	$(F_1, F_1, F_1 + F_2 - M_1 - M_2, F_2)$	20	$(F_1 + \frac{F_2-M_{1+2}}{2}, \frac{F_2+M_1-M_2}{2}, F_1 + \frac{F_2-M_{1+2}}{2}, \frac{F_2+M_1-M_2}{2})$
7	$(F_1 + \frac{F_2-M_{1+2}}{2}, F_1 + \frac{F_2-M_{1+2}}{2}, F_1 + \frac{F_2-M_{1+2}}{2}, \frac{3F_2-M_{1+2}}{2})$	21	$(F_1 + F_2 - M_2, F_1 - M_1, F_1 + F_2 - M_2, F_2)$
8	$(F_1 - M_2, F_1 + F_2 - M_1, F_1 - M_2, F_2)$	22	$(F_1 + F_2 - M_2, F_1 + F_2 - M_1, F_1 + F_2 - M_2, 0)$
9	$(F_1, F_1 + F_2 - M_1 - M_2, F_1, F_2)$	23	$(F_1 + \frac{F_2}{2} - M_2, F_1 + \frac{F_2}{2} - M_1, F_1 + \frac{F_2}{2} - M_2, \frac{F_2}{2})$
10	$(F_1, F_1 + F_2 - M_1, F_1, F_2 - M_2)$	24	$(F_1 + F_2 - M_2, 0, F_1 + F_2 - M_2, F_1 + F_2 - M_1)$
11	$(F_1 - \frac{M_2}{2}, F_1 + F_2 - M_1 - \frac{M_2}{2}, F_1 - \frac{M_2}{2}, F_2 - \frac{M_2}{2})$	25	$(\frac{F_{1+2}+M_1}{2} - M_2, \frac{F_{1+2}-M_1}{2}, \frac{F_{1+2}+M_1}{2} - M_2, \frac{F_{1+2}-M_1}{2})$
12	$(F_1 + F_2 - M_1, F_1 + F_2 - M_1, F_1 - M_2, F_2)$	26	$(0, F_1 + F_2 - M_2, F_1 + F_2 - M_1, F_1 + F_2 - M_2)$
13	$(F_1 + \frac{F_2-M_{1+2}}{2}, F_1 + \frac{F_2-M_{1+2}}{2}, F_1 + \frac{F_2-M_{1+2}}{2}, \frac{F_2+M_1-M_2}{2})$	27	$(F_1 + F_2 - M_1, F_1 + F_2 - M_2, 0, F_1 + F_2 - M_2)$
14	$(F_1 - M_2, F_2, F_1 - M_2, F_2)$	28	$(\frac{F_{1+2}-M_1}{2}, \frac{F_{1+2}+M_1}{2} - M_2, \frac{F_{1+2}-M_1}{2}, \frac{F_{1+2}+M_1}{2} - M_2)$

For example, combining Instances 0 to 2 leads to $R_{(1,2)} \geq \max(0, F_1 - M_1, F_2 - M_2, F_1 + F_2 - M_1 - M_2)$, which is equivalent to (A2) since $M_1 + M_2 \leq F_2$. Inequalities (A1), (A3), and (A4) can be derived similarly. Additionally, we have

$$R_{(1,1)} + R_{(1,2)} \geq 2F_1 + F_2 - M_1 - M_2 \quad \text{(B1)}$$
$$R_{(1,1)} + R_{(2,1)} \geq 2F_1 + F_2 - M_1 - M_2 \quad \text{(B2)}$$
$$R_{(1,1)} + R_{(2,2)} \geq F_1 + F_2 - M_2 \quad \text{(B3)}$$
$$R_{(1,2)} + R_{(2,1)} \geq 2F_1 + F_2 - M_1 - M_2 \quad \text{(B4)}$$
$$R_{(1,2)} + R_{(2,2)} \geq F_1 + 2F_2 - M_1 - M_2 \quad \text{(B5)}$$
$$R_{(2,1)} + R_{(2,2)} \geq F_1 + 2F_2 - M_1 - M_2 \quad \text{(B6)}$$

where the conditions $0 \leq M_2 \leq M_1$ and $M_1 + M_2 \leq F_2$ are used repeatedly to simplify the $\max$ operations that arise when summarizing the 14 inequalities in Instances 3 and 4.

There are 10 inequalities in (A1) to (B6). Each corner point of the 4-dimensional PRCR must satisfy at least 4 of them with equalities. If an inequality is satisfied with equality, we say such an inequality is *tight*. Therefore, we need to have at least 4 tight inequalities. We consider the following 5 cases.

Case 1: (A1) is tight. i.e., $R_{(1,1)} = F_1 - M_2$. We can then combine (A2) and (B1) to obtain

$$R_{(1,2)} \geq \max(F_1 + F_2 - M_1 - M_2, F_1 + F_2 - M_1) \quad (18)$$
$$= F_1 + F_2 - M_1; \quad (19)$$

where (18) follows from substituting $R_{(1,1)} = F_1 - M_2$ into (B1). Similarly, we can combine (A3) and (B2) to obtain

$$R_{(2,1)} \geq \max(F_1 + F_2 - M_1 - M_2, F_1 + F_2 - M_1)$$
$$= F_1 + F_2 - M_1; \quad (20)$$

and combine (A4) and (B3) to obtain

$$R_{(2,2)} \geq \max(F_2 - M_2, F_2) = F_2. \tag{21}$$

We then notice that any $R_{(1,2)}$ and $R_{(2,1)}$ satisfying (19) and (20) automatically satisfy (B4). That is,

$$\begin{aligned} R_{(1,2)} + R_{(2,1)} &\geq 2(F_1 + F_2 - M_1) \\ &\geq 2F_1 + F_2 - M_1 - M_2 \end{aligned} \tag{22}$$

where (22) follows from the condition $M_1 + M_2 \leq F_2$. Similarly, one can show that any $R_{(1,2)}$, $R_{(2,1)}$, and $R_{(2,2)}$ satisfying (19) to (21) automatically satisfy (B4) to (B6).

From the above arguments, the four inequalities that are needed to form the corner point can only be (A1), (19), (20), and (21). The corresponding corner point is thus *Vertex 1* $(F_1 - M_2, F_1 + F_2 - M_1, F_1 + F_2 - M_1, F_2)$.

Cases 2 to 4 follow very similar steps as in Case 1. Specifically, Case 2 considers the case that (A2) is tight, and one can show that the corresponding corner point is indeed Vertex 2. Case 3 considers that (A3) is tight. The corresponding corner point is also[2] Vertex 2. Case 4 considers that (A4) is tight and the corner point is Vertex 3.

Case 5: None of (A1) to (A4) is tight. We first notice that for any corner point satisfying (A1) to (A4) loosely must also satisfy (B4) loosely. That is,

$$\begin{aligned} R_{(1,2)} + R_{(2,1)} &> 2(F_1 + F_2 - M_1 - M_2) \\ &\geq 2F_1 + F_2 - M_1 - M_2 \end{aligned} \tag{23}$$

where (23) follows from the condition $M_1 + M_2 \leq F_2$. Therefore, the corner point $(R_{(1,1)}, R_{(1,2)}, R_{(2,1)}, R_{(2,2)})$ must satisfy 4 out of the 5 equations in (B1) to (B3), (B5), and (B6).

We now argue that the corner point in Case 5 must satisfy $R_{(1,2)} = R_{(2,1)}$. Suppose not, say $R_{(1,2)} > R_{(2,1)}$. By noticing that the right-hand sides of (B1) and (B2) are identical and the left-hand sides share the common $R_{(1,1)}$ term, we know that (B1) must be loose. Similarly, by comparing (B5) and (B6), the assumption $R_{(1,2)} > R_{(2,1)}$ implies (B5) must be loose. Therefore, only 3 equalities (B2), (B3) and (B6) can possibly be tight, which contradicts the fact that each corner point corresponds to at least 4 tight inequalities.

Since $R_{(1,2)} = R_{(2,1)}$, the 5 equations in (B1), (B2), (B3), (B5), and (B6) collapse to three equations (B2), (B3), and (B6) that involve only 3 free variables (since $R_{(1,2)} = R_{(2,1)}$). Solving the three equations leads to a solution *Vertex 4* $(F_1 - \frac{M_2}{2}, F_1 + F_2 - M_1 - \frac{M_2}{2}, F_1 + F_2 - M_1 - \frac{M_2}{2}, F_2 - \frac{M_2}{2})$. By verifying that Vertex 4 also satisfies (A1) to (A4), we have proven that Vertex 4 is the legitimate corner point of Case 5.

All four vertices can be achieved by explicit code construction. Due to space limits, we provide the achievable scheme only for the most interesting case, Vertex 4.

Denote file 1 by U and file 2 by V. We divide file 1 into four disjoint subfiles U_1, U_2, U_3 and U_4 with file size $M_1 - M_2$, $M_2/2$, $M_2/2$, and $F_1 - M_1$ respectively. We divide file 2 into four disjoint subfiles V_1, V_2, V_3, and V_4, with file sizes $M_1 - M_2$, $M_2/2$, $M_2/2$, and $F_2 - M_1$, respectively. Then we have the following zero-error scheme. User 1 caches $Z_1 = (U_1 \oplus V_1, U_2, V_2)$ with total memory size M_1 and user 2 caches $Z_2 = (U_3, V_3)$ with total memory size M_2. In the delivery phase, the transmitted signals for the four demands are $X_{(1,1)} = (U_1, U_2 \oplus U_3, U_4)$, $X_{(1,2)} = (V_1, V_2 \oplus U_3, U_4, V_4)$, $X_{(2,1)} = (U_1, U_2 \oplus V_3, U_4, V_4)$, and $X_{(2,2)} = (V_1, V_2 \oplus V_3, V_4)$. The corresponding rates $R_{(1,1)} = F_1 - \frac{M_2}{2}$, $R_{(1,2)} = F_1 + F_2 - M_1 - \frac{M_2}{2}$, $R_{(2,1)} = F_1 + F_2 - M_1 - \frac{M_2}{2}$, and $R_{(2,2)} = F_2 - \frac{M_2}{2}$ attain the corner point Vertex 4. Q.E.D.

[2]One can prove that for any corner point, if one of its 4 tight inequalities is (A2), then (A3) is also a tight inequality and vice versa. This is why Cases 2 and 3 have the same vertex.

IV. Conclusion

The per-request capacity region (PRCR) is the most fundamental performance metric in the information-theoretic studies of coded caching. In this work, we have characterized the exact PRCR of the 2-user/2-file setting with full heterogeneity and used it to derive the average-rate capacity with heterogeneous demand popularity, file sizes, and cache sizes. The results shed new insights for the future work with general N and K values. For example, the capacity-achieving schemes reported in [15] are all based on space-sharing among 7 basic schemes, which are likely to be useful for general N and K as well.

References

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

[2] C. Tian, "Symmetry, demand types and outer bounds in caching systems," in *Proc. IEEE Int. Symp. Inform. Theory (ISIT)*, Jul. 2016, pp. 825–829.

[3] D. Cao, D. Zhang, P. Chen, N. Liu, W. Kang, and D. Gunduz, "Coded caching with heterogeneous cache sizes and link qualities: The two-user case," in *Proc. IEEE Int. Symp. Inform. Theory (ISIT)*, Jun. 2018, pp. 1545–1549.

[4] M. A. Maddah-Ali and U. Niesen, "Decentralized coded caching attains order-optimal memory-rate tradeoff," *IEEE/ACM Trans. Netw.*, vol. 23, no. 4, pp. 1029–1040, Aug 2015.

[5] U. Niesen and M. A. Maddah-Ali, "Coded caching with nonuniform demands," *IEEE Trans. Inf. Theory*, vol. 63, no. 2, pp. 1146–1158, Feb 2017.

[6] J. Zhang, X. Lin, and X. Wang, "Coded caching under arbitrary popularity distributions," *IEEE Trans. Inf. Theory*, vol. 64, no. 1, pp. 349–366, Jan. 2018.

[7] M. Ji, A. M. Tulino, J. Llorca, and G. Caire, "Order-optimal rate of caching and coded multicasting with random demands," *IEEE Trans. Inf. Theory*, vol. 63, no. 6, pp. 3923–3949, Jun. 2017.

[8] 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. Inf. Theory*, vol. 65, no. 1, pp. 647–663, Jan 2019.

[9] C. Li, "On rate region of caching problems with non-uniform file and cache sizes," *IEEE Commun. Lett.*, vol. 21, no. 2, pp. 238–241, Feb 2017.

[10] A. M. Daniel and W. Yu, "Optimization of heterogeneous coded caching," *arXiv:1708.04322*, Aug. 2017.

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

[12] P. Quinton, S. Sahraei, and M. Gastpar, "A novel centralized strategy for coded caching with non-uniform demands," *arXiv:1801.10563*, 2018.

[13] J. Zhang, X. Lin, C. Wang, and X. Wang, "Coded caching for files with distinct file sizes," in *Proc. IEEE Int. Symp. Inform. Theory (ISIT)*, June 2015, pp. 1686–1690.

[14] H. Ghasemi and A. Ramamoorthy, "Improved lower bounds for coded caching," *IEEE Trans. Inf. Theory*, vol. 63, no. 7, pp. 4388–4413, July 2017.

[15] https://engineering.purdue.edu/~chihw/pub_pdf/ISIT2019_micro_cap_proof.pdf.

[16] A. M. Ibrahim, A. A. Zewail, and A. Yener, "Optimization of heterogeneous caching systems with rate limited links," in *Proc. IEEE Int. Conf. on Communications (ICC)*, May 2017, pp. 1–6.