

Memory-Rate Tradeoff for Decentralized Caching under Nonuniform File Popularity

Yong Deng and Min Dong

Dept. of Electrical, Computer and Software Engineering, Ontario Tech University, Ontario, Canada

Abstract—We study the memory-rate tradeoff for decentralized caching under nonuniform file popularity. We formulate the cache placement optimization problem for a recently proposed decentralized modified coded caching scheme (D-MCCS) to minimize the average rate. To solve this non-convex optimization problem, we develop two algorithms: a successive Geometric Programming (GP) approximation algorithm, which guarantees convergence to a stationary point but has a high computational complexity, and a low-complexity approach based on a two-file-group-based placement strategy. We further propose a lower bound on the average rate for decentralized caching under nonuniform file popularity. The lower bound is given as a non-convex optimization problem, for which we propose a similar successive GP approximation algorithm to compute a stationary point. We show that the optimized MCCS attains the lower bound for the special case of no more than two active users requesting files, or for the general case but satisfying a special condition. Thus, the optimized MCCS characterizes the exact memory-rate tradeoff for decentralized caching in these cases. In general, our numerical result shows that the optimized D-MCCS performs close to the lower bound.

I. INTRODUCTION

Caching is anticipated to be a key technique to address the pressing issues of network congestion and content delivery delay [1]. To maximize the caching gain, the authors of [2] proposed a coded caching scheme that combines an uncoded placement and a coded multicast delivery strategy to explore the global caching gain. The scheme has been shown to substantially reduce the delivery rate as compared with uncoded caching. This promising result has attracted extensive studies on coded caching under different systems or network structures [3]–[10].

The above studies generally rely on a carefully designed centrally coordinated cache placement strategy to store a portion of each file content in a subset of users. However, a coordinated cache placement may not always be possible in practice, which may limit the practical use of coded caching. For this issue, decentralized caching has been considered [11], where no coordination among users is required, and each user caches uncoded contents independently from each other. Specifically, for a system with a central server connecting to multiple cache-equipped users, a *decentralized coded caching scheme* (D-CCS) was proposed in [11], which consists of a decentralized (uncoded) placement scheme and a coded delivery strategy. Interestingly, under uniform file popularity, it is shown that the performance of the D-CCS is close to the centralized coded caching scheme [11]. The D-CCS has since attracted many interests, with extensions to nonuniform

cache sizes [12], [13], and nonuniform file popularity [14]–[16] or sizes [17]. For nonuniform file popularity, existing works mainly focus on the cache placement strategies for the D-CCS [14]–[17] to reduce the achievable rates. To quantify the performances of these proposed schemes for D-CCS, [14]–[17] proposed different lower bounds on the average rate for caching with any placement. With the number of users requesting files (*i.e.*, active users) known at the server, it has been shown that the achievable rate of the D-CCS is within a factor away from the tightest lower bound developed in [16]. However, since the lower bound is for any caching, the gap is still large for practical consideration. These existing results [14]–[17] are both not sufficient to characterize the memory-rate tradeoff for decentralized caching, especially for the case when the users who request files are unknown to the server.

Recently, for files with uniform popularity, a *decentralized modified coded caching scheme* (D-MCCS) was proposed in [18] that removes the redundancy in the coded messages used in the D-CCS to further reduce the delivery rate. It has been further shown that the D-MCCS attains the lower bounds on both average and peak rates for decentralized caching and thus characterizes the exact memory-rate tradeoff [18]. For files with nonuniform popularity, there is no study on the cache placement optimization for the D-MCCS or how optimal the D-MCCS is for decentralized caching. In general, the memory-rate tradeoff for decentralized caching remains unknown.

In this paper, we aim at characterizing the memory-rate tradeoff in terms of the average rate for decentralized caching under nonuniform file popularity. In particular, for the D-MCCS, we formulate the cache placement optimization problem to minimize the average rate. To solve this challenging non-convex optimization problem, we first propose a successive Geometric Programming (GP) approximation algorithm, which guarantees convergence to a stationary point. Due to the high computational complexity of this algorithm, we further develop a low-complexity approximate approach by using a two-file-group-based cache placement strategy. Both algorithms assume unknown active users at a given time. Our numerical result shows that both algorithms perform close to each other. We then propose a lower bound on the average rate for decentralized caching under nonuniform file popularity. The lower bound is presented as a non-convex optimization problem, and we develop a similar successive GP approximation algorithm that converges to a stationary point. For the special case of no more than two active users requesting files, we show that the optimized D-MCCS attains the lower bound and thus is an optimal decentralized caching

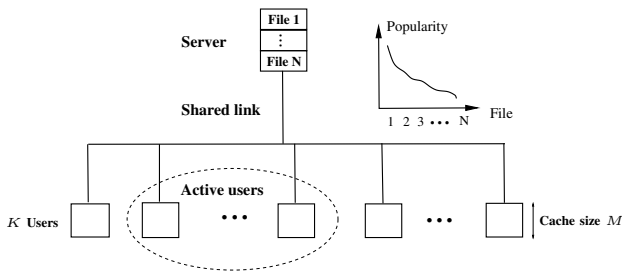


Fig. 1. A cache-aided system with end users each equipped with a local cache connecting to the server via a shared link. The files in the server have nonuniform popularity. Only a number of unknown active users request files during the delivery phase.

scheme that characterizes the exact memory-rate tradeoff for decentralized caching. For the general case, we also identify a special condition for the optimized D-MCCS to attain the lower bound. Numerical results show that the performance gap between the optimized D-MCCS and the lower bound is very small in general, implying that the optimized D-MCCS achieves close-to-optimal performance for decentralized caching under nonuniform file popularity.

II. SYSTEM MODEL

We consider a cache-aided transmission system where a server connects to K users over a shared error-free link, as shown in Fig 1. We denote the user index set by $\mathcal{K} \triangleq \{1, \dots, K\}$. Each user $k \in \mathcal{K}$ has a local cache of capacity MF bits, which is referred to as cache size M (normalized by the file size). The server has a database consisting of N files, denoted as W_1, \dots, W_N . Denote the file index set by $\mathcal{N} \triangleq \{1, \dots, N\}$. Each file $W_n, n \in \mathcal{N}$, is of size F bits and has probability p_n of being requested. We denote $\mathbf{p} \triangleq [p_1, \dots, p_N]$ as the popularity distribution of all N files, where $\sum_{n=1}^N p_n = 1$. Without loss the generality, we label files according to the decreasing order of their popularity as $p_1 \geq p_2 \geq \dots \geq p_N$.

The caching scheme operates in two phases: the cache placement phase and the content delivery phase. In the cache placement phase, all users have access to the files stored in the server. For each file $n \in \mathcal{N}$, the users select a portion of its uncoded contents to store into their local caches. With decentralized caching, the cached contents are selected randomly by each user without any coordination. In the content delivery phase, a subset of users in \mathcal{K} request files from the server. We refer to these users who are requesting files as active users. Note that the identities of the active users are unknown to the server in the cache placement phase. Let $p_{a,k}$ denote the probability of user k being active in the delivery phase. We define $\mathbf{p}_a \triangleq [p_{a,1}, \dots, p_{a,K}]$ as the probability vector of users being active. Let $\mathcal{A} \subseteq \mathcal{K}$ denote the active user set. Let d_k denote the index of the file requested by active user $k \in \mathcal{A}$. We define the demand vector of all the active users in \mathcal{A} as $\mathbf{d}_A \triangleq (d_k)_{k \in \mathcal{A}}$. Based on the demand vector \mathbf{d}_A and the cached contents at users, the server generates coded messages containing the uncached portion of requested files and transmits them to the active users. Upon receiving the coded messages, each active user $k \in \mathcal{A}$ reconstructs its

requested file $\hat{W}_{\mathbf{d}_A, k}$ from the received coded messages and its cached content. For a valid coded caching scheme, each active user $k \in \mathcal{A}$ is able to reconstruct its requested file, $\hat{W}_{\mathbf{d}_A, k} = W_{d_k}$, for any demand vector \mathbf{d}_A , assuming an error-free link.

III. DECENTRALIZED MODIFIED CODED CACHING SCHEME

In this section, we describe the cache placement and content delivery procedures of the D-MCCS under nonuniform file popularity.

A. Decentralized Cache Placement

A salient feature of decentralized caching is that the active user set \mathcal{A} (both the size and the user identities) is unknown during the cache placement phase. We consider the following decentralized placement procedure: each user $k \in \mathcal{K}$ independently and randomly selects and caches $q_n F$ bits of file $W_n, n \in \mathcal{N}$, where q_n is the portion of the file the user wants to cache, *i.e.*,

$$0 \leq q_n \leq 1, \quad n \in \mathcal{N}. \quad (1)$$

We define $\mathbf{q} \triangleq [q_1, \dots, q_N]$ as the cache placement vector of all the files in \mathcal{N} . For uniform file popularity, the symmetrical decentralized placement is optimal for the D-MCCS [18], *i.e.*, $q_1 = \dots = q_N = \frac{M}{N}$. For nonuniform file popularity, the cache placement may be different for different files, which complicates the cache placement design of the D-MCCS. In this work, we aim to optimize the cache placement vector \mathbf{q} for the D-MCCS to minimize the average delivery rate. Since a subset of file $n \in \mathcal{N}$ of $q_n F$ bits are cached by each user of cache size M , we have the cache size constraint given by

$$\sum_{n=1}^N q_n F \leq MF. \quad (2)$$

Note that the server knows the cached contents by each user.

B. Content Delivery

During the delivery process, the server receives the information of the active user set \mathcal{A} and their demand vector \mathbf{d}_A . Based on these, the server knows the cached contents among the users in \mathcal{A} . We define subfile $W_{n,S}$ as the chunk of file W_n that is cached by the active user subset $\mathcal{S} \subseteq \mathcal{A}$ but not by the rest users in \mathcal{A} , *i.e.*, $\mathcal{A} \setminus \mathcal{S}$. We use $W_{n,\emptyset}$ to represent the portion of file n that is not cached by any user in \mathcal{A} . Under the decentralized placement, for file size F being sufficiently large, by the law of large numbers, q_n is approximately the probability of one bit in file n being selected by a user. Following this, the size of subfile $W_{n,S}$ is approximately given by [11]

$$|W_{n,S}| \approx q_n^s (1 - q_n)^{A-s} F, \quad \mathcal{S} \subseteq \mathcal{A}, |\mathcal{S}| = s \quad (3)$$

where $A \triangleq |\mathcal{A}|$. According to (3), the size of subfile $W_{n,S}$ depends on $|\mathcal{S}|$, *i.e.*, the number of the users who cache it.

For any file demand vector \mathbf{d}_A , the D-MCCS multicasts coded messages to different user subsets in \mathcal{A} . Each coded

message is intended for an unique active user subset $\mathcal{S} \subseteq \mathcal{A}$ and is formed by the bitwise XOR operation of total $|\mathcal{S}|$ subfiles, one from each requested file, given by

$$C_{\mathcal{S}} \triangleq \bigoplus_{k \in \mathcal{S}} W_{d_k, \mathcal{S} \setminus \{k\}}, \quad \mathcal{S} \subseteq \mathcal{A}, \mathcal{S} \neq \emptyset. \quad (4)$$

From (4), each of the subfiles in $C_{\mathcal{S}}$ belongs to file d_k requested by user $k \in \mathcal{S}$ and is cached by users in $\mathcal{S} \setminus \{k\}$ exclusively. Note that the coded messages can only be formed for the nonempty active user subset $\mathcal{S} \neq \emptyset$.

Since the portion q_n of file n cached by the users may be different for files with different popularities, the subfiles forming the coded message $C_{\mathcal{S}}$ in (4) may not have equal size. In this case, zero-padding is adopted for the XOR operation such that subfiles are zero-padded to the size of the longest subfile. Thus, the size of $C_{\mathcal{S}}$ is determined by the largest subfile in $C_{\mathcal{S}}$, *i.e.*,

$$|C_{\mathcal{S}}| = \max_{k \in \mathcal{S}} |W_{d_k, \mathcal{S} \setminus \{k\}}| = \max_{k \in \mathcal{S}} q_{d_k}^s (1 - q_{d_k})^{A-s} F, \\ \mathcal{S} \subseteq \mathcal{A}, |\mathcal{S}| = s + 1, s = 0, \dots, A - 1. \quad (5)$$

Remark 1. For nonuniform file popularity, cache placement may be different for different files, resulting in generating subfiles of nonequal sizes. The existence of nonequal subfiles complicates the cache placement design. Zero-padding is a common technique to handle the nonequal files in formulating the coded messages for both centralized [4]–[6] and decentralized coded caching [17]. However, its impact on decentralized coded caching has never been studied and is unknown. In Section V-B, by developing a matching converse bound, we show that there is no loss of optimality by using zero-padding in decentralized coded caching if the size of the active user set is no more than two, $A \leq 2$.

In the original D-CCS [11], for any file demand vector \mathbf{d}_A , the server transmits the coded messages corresponding to all the active user subsets $\{C_{\mathcal{S}} : \forall \mathcal{S} \subseteq \mathcal{A}\}$ to the active users. For the D-MCCS, the server only transmits coded messages corresponding to certain selected active user subsets [18]. We first provide the following two definitions:

Definition 1. Leader group: For any demand vector \mathbf{d}_A with $\tilde{N}(\mathbf{d}_A)$ distinct requests, the leader group \mathcal{U}_A is a subset of the active user set \mathcal{A} , *i.e.*, $\mathcal{U}_A \subseteq \mathcal{A}$, that satisfies $|\mathcal{U}_A| = \tilde{N}(\mathbf{d}_A)$ and the users in \mathcal{U}_A have exactly $\tilde{N}(\mathbf{d}_A)$ distinct requests.

Definition 2. Redundant group: Given \mathcal{U}_A , any active user subset $\mathcal{S} \subseteq \mathcal{A}$ is called a redundant group if $\mathcal{S} \cap \mathcal{U}_A = \emptyset$; otherwise, \mathcal{S} is a non-redundant group.

The delivery procedure of the D-MCCS improves upon that of the D-CCS by only multicasting coded messages corresponding to the non-redundant groups, *i.e.*, $\{C_{\mathcal{S}} : \forall \mathcal{S} \subseteq \mathcal{A} \text{ and } \mathcal{S} \cap \mathcal{U}_A \neq \emptyset\}$, to both non-redundant and redundant groups.¹ As a result, the D-MCCS achieves a lower delivery rate than the D-CCS. Note that the rate reduction only occurs when

¹Note that this coded delivery strategy follows that of the centralized MCCS [18], which has been shown to be a valid scheme, *i.e.*, a user can reconstruct any requested file.

Algorithm 1 Decentralized modified coded caching scheme

Decentralized cache placement procedure:

- 1: **for** $n \in \mathcal{N}$ **do**
- 2: Each user randomly caches $q_n F$ bit of file W_n .
- 3: **end for**

Coded delivery procedure:

- 1: **for** $\mathcal{S} \subseteq \mathcal{A}$ and $\mathcal{S} \cap \mathcal{U}_A \neq \emptyset$ **do**
 - 2: Server generates $C_{\mathcal{S}}$ and multicasts it to \mathcal{S} .
 - 3: **end for**
-

there exist redundant groups, *i.e.*, there are multiple requests of the same file among the active users.

We summarize both the cache placement and coded delivery procedures of the D-MCCS in Algorithm 1. With the cached contents at each user via the decentralized cache placement described in Section III-A and the coded messages $\{C_{\mathcal{S}} : \forall \mathcal{S} \subseteq \mathcal{A} \text{ and } \mathcal{S} \cap \mathcal{U}_A \neq \emptyset\}$ multicasted by the server, each user in \mathcal{A} can retrieve all the subfiles of its requested file [18].

IV. DECENTRALIZED CACHE PLACEMENT OPTIMIZATION

In this section, we first formulate the cache placement design for the D-MCCS under nonuniform file popularity as an optimization problem to minimize the average delivery rate. We then develop two algorithms to solve the problem.

A. Problem Formulation

Based on the delivery procedure in the D-MCCS described in Section III-B, for a given demand vector \mathbf{d}_A , the delivery rate is the total number of bits in the coded messages corresponding to all the non-redundant groups $\{C_{\mathcal{S}} : \forall \mathcal{S} \subseteq \mathcal{A} \text{ and } \mathcal{S} \cap \mathcal{U}_A \neq \emptyset\}$, expressed as

$$R_{\text{MCCS}}(\mathbf{d}_A; \mathbf{q}) = \sum_{\mathcal{S} \subseteq \mathcal{A}, \mathcal{S} \cap \mathcal{U}_A \neq \emptyset} |C_{\mathcal{S}}|. \quad (6)$$

Define $\mathcal{Q}^s \triangleq \{\mathcal{S} \subseteq \mathcal{A} : \mathcal{S} \cap \mathcal{U}_A = \emptyset, |\mathcal{S}| = s\}$ as the set of the non-redundant groups with $|\mathcal{S}| = s$ users for $s = 1, \dots, K$. Based on (5), we can rewrite (6) as

$$R_{\text{MCCS}}(\mathbf{d}_A; \mathbf{q}) = \sum_{s=0}^{A-1} \sum_{\mathcal{S} \in \mathcal{Q}^{s+1}} \max_{k \in \mathcal{S}} q_{d_k}^s (1 - q_{d_k})^{A-s} F. \quad (7)$$

By taking the expectation of $R_{\text{MCCS}}(\mathbf{d}_A; \mathbf{q})$ over all the possible $\mathbf{d}_A \in \mathcal{N}^A$ and $\mathcal{A} \subseteq \mathcal{K}$, the average rate of the D-MCCS as a function of \mathbf{q} is given by

$$\bar{R}_{\text{MCCS}}(\mathbf{q}) = E_{\mathcal{A}} [E_{\mathbf{d}_A} [R_{\text{MCCS}}(\mathbf{d}_A; \mathbf{q})]] \\ = E_{\mathcal{A}} \left[\sum_{\mathbf{d}_A \in \mathcal{N}^A} \left(\prod_{k \in \mathcal{A}} p_{d_k} \right) R_{\text{MCCS}}(\mathbf{d}_A; \mathbf{q}) \right]. \quad (8)$$

Thus, we formulate the cache placement optimization problem for the D-MCCS under nonuniform file popularity as

$$\mathbf{P0} : \min_{\mathbf{q}} \bar{R}_{\text{MCCS}}(\mathbf{q}) \quad \text{s.t.} \quad (1), (2).$$

P0 is a non-convex optimization problem w.r.t. \mathbf{q} , which is difficult to solve. In the following subsection, we propose two different algorithms to solve **P0**.

B. Optimal Decentralized Cache Placement Solutions

We first develop an algorithm that converges to a stationary point of $\mathbf{P0}$ through solving a series of Geometric Programming (GP) problems. To reduce the computational complexity, we further propose an approximate solution with very low complexity to compute.

1) *Successive GP Approximation Algorithm:* We reformulate $\mathbf{P0}$ into an equivalent Complementary GP (CGP) problem, which is an intractable NP-hard problem [19]. It is shown that a stationary point of a CGP can be obtained through the generic successive approximation method [20].

To reformulate $\mathbf{P0}$ into an equivalent CGP problem, we first introduce auxiliary variables $x_n, n \in \mathcal{N}$, and add the following inequality constraint for $(1 - q_n)$ in (7).

$$1 - q_n \leq x_n, \quad n \in \mathcal{N}. \quad (9)$$

We also introduce auxiliary variables $w_{\mathbf{d}_A, \mathcal{S}}$ for $\mathcal{A} \subseteq \mathcal{K}, \mathbf{d}_A \in \mathcal{N}^A$ and $\mathcal{S} \in \mathcal{Q}^{s+1}, s = 0, \dots, K-1$. By (9), we replace $\max_{k \in \mathcal{S}} q_{d_k}^s (1 - q_{d_k})^{A-s} F$ in (7) with $w_{\mathbf{d}_A, \mathcal{S}}$ and add the following constraints

$$q_{d_k}^s x_{d_k}^{A-s} F \leq w_{\mathbf{d}_A, \mathcal{S}}, \quad k \in \mathcal{S} \quad (10)$$

for given $\mathcal{S} \in \mathcal{Q}^{s+1}, \mathbf{d}_A \subseteq \mathcal{N}^A, \mathcal{A} \subseteq \mathcal{K}$. As a result, we reformulate $\mathbf{P0}$ into the following equivalent problem

$$\mathbf{P1} : \min_{\mathbf{q}, \mathbf{x}, \mathbf{w} \geq 0} E_{\mathcal{A}} \left[\sum_{\mathbf{d}_A \in \mathcal{N}^A} \left(\prod_{k \in \mathcal{A}} p_{d_k} \right) \sum_{s=0}^{A-1} \sum_{\mathcal{S} \in \mathcal{Q}^{s+1}} w_{\mathbf{d}_A, \mathcal{S}} \right] \quad (11)$$

$$\text{s.t. } q_n \leq 1, \quad n \in \mathcal{N}, \quad (12)$$

$$\sum_{n=1}^N q_n M^{-1} \leq 1, \quad (12)$$

$$\frac{1}{q_n + x_n} \leq 1, \quad n \in \mathcal{N}, \quad (13)$$

$$w_{\mathbf{d}_A, \mathcal{S}}^{-1} \cdot q_{d_k}^s x_{d_k}^{A-s} F \leq 1, \quad k \in \mathcal{S}, \mathcal{S} \in \mathcal{Q}^{s+1}, \quad (14)$$

where $\mathbf{x} \triangleq (x_n)_{n \in \mathcal{N}}$ and $\mathbf{w} \triangleq (w_{\mathbf{d}_A, \mathcal{S}})_{\mathcal{S} \in \mathcal{Q}^{s+1}, \mathbf{d}_A \subseteq \mathcal{N}^A, \mathcal{A} \subseteq \mathcal{K}}$. Note that constraints (12), (13) and (14) are direct reformulations of (2), (9) and (10), respectively. $\mathbf{P1}$ minimizes a posynomial subject to upper bound three inequality constraints (11), (12) and (14) that are posynomials and the inequality constraint (13) that is on the ratio between two posynomials. Thus, $\mathbf{P1}$ is a CGP problem. For a CGP problem, an approach was developed using a sequence of approximate GPs to obtain a stationary point of the problem [20]. We adopt this approach to solve $\mathbf{P1}$ by solving $(\mathbf{q}, \mathbf{x}, \mathbf{w})$ iteratively via a sequence of approximate GP problems. Define the objective function of $\mathbf{P1}$ by $\bar{R}_{\text{MCCS}}^{\text{CGP}}$. In the i th iteration, given $(\mathbf{q}^{(i)}, \mathbf{x}^{(i)})$ obtained from previous iteration, we have the following approximate GP problem of $\mathbf{P1}$.

$$\mathbf{P2}(\mathbf{q}^{(i)}, \mathbf{x}^{(i)}) : (\mathbf{q}^{(i+1)}, \mathbf{x}^{(i+1)}, \mathbf{w}^{(i+1)}) = \underset{\mathbf{q}, \mathbf{x}, \mathbf{w} \geq 0}{\text{argmin}} \bar{R}_{\text{MCCS}}^{\text{CGP}}(\mathbf{q}, \mathbf{x}, \mathbf{w})$$

$$\text{s.t. (11), (12), (14),}$$

$$\frac{1}{\left(q_n^{(i)} + x_n^{(i)} \right) \left(\frac{q_n}{q_n^{(i)}} \right)^{\alpha_n^{(i)}} \left(\frac{x_n}{x_n^{(i)}} \right)^{\beta_n^{(i)}}} \leq 1, \quad n \in \mathcal{N} \quad (15)$$

where $\alpha_n^{(i)} \triangleq \frac{q_n^{(i)}}{q_n^{(i)} + x_n^{(i)}}$ and $\beta_n^{(i)} \triangleq \frac{x_n^{(i)}}{q_n^{(i)} + x_n^{(i)}}$. Note that constraint (15) is formed using (13) and the arithmetic-geometric mean inequality [20]

$$q_n + x_n \geq \left(\frac{q_n}{\alpha_n^{(i)}} \right) \left(\frac{x_n}{\beta_n^{(i)}} \right) \geq 1. \quad (16)$$

Note that $\mathbf{P2}(\mathbf{q}^{(i)}, \mathbf{x}^{(i)})$ is a standard GP problem, which can be solved using a standard convex optimization solver. The above approach of iteratively solving $\mathbf{P2}(\mathbf{q}^{(i)}, \mathbf{x}^{(i)})$ is guaranteed to converge to a stationary point of $\mathbf{P1}$ [20]. This successive GP approximation algorithm is summarized in Algorithm 2. By the equivalence of $\mathbf{P0}$ and $\mathbf{P1}$, we can compute a stationary point of $\mathbf{P0}$ using Algorithm 2. Note that as the size of $\mathbf{P2}(\mathbf{q}^{(i)}, \mathbf{x}^{(i)})$ grows exponentially with K , the computational complexity of Algorithm 2 can be very high. To address this, next, we develop an alternative algorithm to solve the problem with very low complexity.

Algorithm 2 The successive GP approximation algorithm for $\mathbf{P1}$

Input: $K, M, N, \mathbf{p}, \mathbf{p}_a$.

1: **Initialization:** Choose initial feasible point $(\mathbf{q}^{(0)}, \mathbf{x}^{(0)}, \mathbf{w}^{(0)})$. Set $i = 0$.

2: **repeat**

3: Solve $\mathbf{P2}(\mathbf{q}^{(i)}, \mathbf{x}^{(i)})$ to obtain $(\mathbf{q}^{(i+1)}, \mathbf{x}^{(i+1)}, \mathbf{w}^{(i+1)})$.

4: Set $i = i + 1$.

5: **until** $\bar{R}_{\text{MCCS}}^{\text{CGP}}(\mathbf{q}^{(i)}, \mathbf{x}^{(i)}, \mathbf{w}^{(i)})$ converges.

6: $\bar{R}_{\text{MCCS}}^* = \bar{R}_{\text{MCCS}}^{\text{CGP}}(\mathbf{q}^{(i)}, \mathbf{x}^{(i)}, \mathbf{w}^{(i)})$; $\mathbf{q}^* = \mathbf{q}^{(i)}$.

Output: $\bar{R}_{\text{MCCS}}^*, \mathbf{q}^*$.

2) *Low-Complexity File-Group-Based Approach:* We now propose an algorithm that computes an approximate solution for $\mathbf{P0}$. The algorithm is based on a particular cache placement structure. In particular, we consider the two-file-group based placement scheme below.

Two-file-group-based placement: In the cache placement phase, the N files are partitioned into two groups – based on their popularity distribution \mathbf{p} . Define $\mathcal{N}_1 \triangleq \{1, \dots, N_1\}$, for $N_1 \in \mathcal{N}$, and $\mathcal{N}_2 \triangleq \mathcal{N} \setminus \mathcal{N}_1$ as the file index sets of first and second file groups, respectively. The first group \mathcal{N}_1 contains more popular files. For each user $k \in \mathcal{K}$, its entire cache is allocated to the first group \mathcal{N}_1 and is equally split among these files in \mathcal{N}_1 . Thus, the cached portion of each file in the two-file-group based placement is given by

$$q_n = \begin{cases} M/N_1, & n \in \mathcal{N}_1, \\ 0, & n \in \mathcal{N}_2. \end{cases} \quad (17)$$

Let \mathcal{A}_1 and \mathcal{A}_2 denote the sets of active users who request the files in \mathcal{N}_1 and \mathcal{N}_2 , respectively. Note that $\mathcal{A}_1 \cap \mathcal{A}_2 = \emptyset$ and $\mathcal{A} = \mathcal{A}_1 \cup \mathcal{A}_2$. Denote $A_i = |\mathcal{A}_i|$, for $i = 1, 2$. Accordingly, the number of distinct file requests from \mathcal{A}_i is $\tilde{N}(\mathbf{d}_{\mathcal{A}_i})$. Note that \mathcal{A}_i and $\tilde{N}(\mathbf{d}_{\mathcal{A}_i})$, $i = 1, 2$ are all functions of N_1 . Under this two-file-group-based placement structure, in the following lemma, $\mathbf{P0}$ can be reformulated as an optimization problem w.r.t. N_1 to minimize the average rate.

Algorithm 3 Two-file-group based approximate solution**Input:** $K, M, N, \mathbf{p}, \mathbf{p}_a$.1: **for** $N_1 = 1$ to N **do**2: Compute $\bar{R}_{\text{MCCS}}^{\text{FG-2}}(N_1)$ by (18).3: **end for**4: Compute $N_1^* = \text{argmin}_{N_1} \bar{R}_{\text{MCCS}}^{\text{FG-2}}(N_1)$.5: Compute $\bar{R}_{\text{MCCS}}^{\text{FG-2}}(N_1^*)$.**Output:** $\bar{R}_{\text{MCCS}}^{\text{FG-2}}(N_1^*)$.

Proposition 1. Consider the decentralized caching problem of N files with popularity distribution \mathbf{p} and K users, where each user k has cache size M and is with probability $p_{a,k}$ of being active. The minimum average rate under the two-file-group-based decentralized cache placement (17) for the D-MCCS is $\min_{N_1 \in \mathcal{N}} \bar{R}_{\text{MCCS}}^{\text{FG-2}}(N_1)$, where $\bar{R}_{\text{MCCS}}^{\text{FG-2}}(N_1)$ is given by

$$\bar{R}_{\text{MCCS}}^{\text{FG-2}}(N_1) \triangleq E_{\mathcal{A}} \left[\sum_{\mathbf{d}_{\mathcal{A}} \in \mathcal{N}^{\mathcal{A}}} \left(\prod_{k \in \mathcal{A}} p_{d_k} \right) R_{\text{MCCS}}^{\text{FG-2}}(\mathbf{d}_{\mathcal{A}}; N_1) \right] \quad (18)$$

where $R_{\text{MCCS}}^{\text{FG-2}}(\mathbf{d}_{\mathcal{A}}; N_1)$ is the delivery rate under two-file-group-based placement given $\mathbf{d}_{\mathcal{A}}$ and N_1 , expressed as

$$\begin{aligned} R_{\text{MCCS}}^{\text{FG-2}}(\mathbf{d}_{\mathcal{A}}; N_1) = & \\ & \sum_{s=0}^{A-1} \left(\sum_{i=1}^{\tilde{N}(\mathbf{d}_{\mathcal{A}})} \binom{A-i}{s} - \sum_{i=1}^{\tilde{N}(\mathbf{d}_{\mathcal{A}_2})} \binom{A_2-i}{s} \right) \\ & \cdot \left(\frac{M}{N_1} \right)^s \left(1 - \frac{M}{N_1} \right)^{A-s} F + \tilde{N}(\mathbf{d}_{\mathcal{A}_2}) F. \quad (19) \end{aligned}$$

Proof: We omit the detail of the proof due to space limitation. Briefly, by (17), the files in each of the two groups has the same placement. We categorize the coded messages $\{\mathcal{C}_S\}$ into two types, depending on whether a non-redundant group contains users in \mathcal{A}_1 or not. Then, we derive the size $|\mathcal{C}_S|$ in each type and obtain $R_{\text{MCCS}}^{\text{FG-2}}(\mathbf{d}_{\mathcal{A}}; N_1)$ in (19). ■

By Proposition 1, under the two-file-group-based placement, the optimal $N_1 \in \mathcal{N}$ that results in the minimum average rate can be obtained through search in \mathcal{N} . Our algorithm is summarized in Algorithm 3. For each $N_1 \in \mathcal{N}$, the average rate is computed directly using the objective function in (18) using the closed-form expression in (19). Thus, the computational complexity of Algorithm 3 is much lower as compared with Algorithm 2. Interestingly, our numerical study in Section VI shows that the average rate achieved by Algorithm 3 is very close to that of Algorithm 2 and, in some cases, could be even lower than that of Algorithm 2.

Remark 2. Assuming the active user set \mathcal{A} is known, a similar two-file-group-based placement has been considered in [15] [16] for the D-CCS, where the size of the first group N_1 was proposed through heuristics. Our work is different from them in the following aspects: First, we develop our placement solution for the case of unknown active user set \mathcal{A} . Second, the D-MCCS is different from the D-CCS considered in [15], [16] in terms of the delivery procedure. Specifically, the delivery procedure of the D-MCCS removes the redundancy that exists in the coded messages of the D-CCS. Furthermore, [15],

[16] apply a user-grouping-based coded message generation method, where the coded message in (4) is formed by files within the same file group, and there is no coding across file groups. In contrast, we explore the coded caching gain among all the requested files in Algorithm 1. Note that as it has been shown for the D-CCS, the average rate of the coded delivery that explores the coded caching gain among all files is a lower bound to that of the user-grouping-based delivery [10].

V. MEMORY RATE TRADEOFF FOR DECENTRALIZED CACHING

In this section, we characterize the memory-rate tradeoff for decentralized caching under nonuniform file popularity by proposing a lower bound and comparing it with the average rate of the optimized D-MCCS in **P0**.

A. Lower Bound for Decentralized Caching

The general idea for developing the lower bound for decentralized caching is to divide all the possible file demand vectors into different types and then derive a lower bound for each type separately [18]. Given any active user set $\mathcal{A} \subseteq \mathcal{K}$, we categorize all the possible demand vectors $\mathbf{d}_{\mathcal{A}} \in \mathcal{N}^{\mathcal{A}}$ based on the distinct file requests in $\mathbf{d}_{\mathcal{A}}$. We use $\text{Unique}(\mathbf{d}_{\mathcal{A}})$ to denote extracting the distinct file requests in $\mathbf{d}_{\mathcal{A}}$, and the resulting index set of distinct files is denoted as $\mathcal{D}_{\mathcal{A}} \triangleq \text{Unique}(\mathbf{d}_{\mathcal{A}})$. Recall that the leader group $\mathcal{U}_{\mathcal{A}}$ contains $\tilde{N}(\mathbf{d}_{\mathcal{A}})$ users requesting all the distinct files in $\mathbf{d}_{\mathcal{A}}$ and thus we have $|\mathcal{D}_{\mathcal{A}}| = |\mathcal{U}_{\mathcal{A}}| = \tilde{N}(\mathbf{d}_{\mathcal{A}})$.

We present a lower bound on the average rate for decentralized caching under nonuniform file popularity in the following theorem.

Theorem 1. Consider the decentralized caching problem of N files with popularity distribution \mathbf{p} and K users, where each user k has cache size M and is with probability $p_{a,k}$ of being active. The following optimization problem provides a lower bound on the average rate:

$$\begin{aligned} \mathbf{P3}: \min_{\mathbf{q}} \bar{R}_{\text{lb}}(\mathbf{q}) \triangleq E_{\mathcal{A}} \left[\sum_{\mathcal{D}_{\mathcal{A}} \subseteq \mathcal{N}} \sum_{\mathbf{d}_{\mathcal{A}} \in \mathcal{T}(\mathcal{D}_{\mathcal{A}})} \left(\prod_{k \in \mathcal{A}} p_{d_k} \right) R_{\text{lb}}(\mathcal{D}_{\mathcal{A}}; \mathbf{q}) \right] \quad (20) \\ \text{s.t. (1), (2)} \end{aligned}$$

where $\mathcal{T}(\mathcal{D}_{\mathcal{A}}) \triangleq \{\mathbf{d}_{\mathcal{A}} : \text{Unique}(\mathbf{d}_{\mathcal{A}}) = \mathcal{D}_{\mathcal{A}}, \mathbf{d}_{\mathcal{A}} \in \mathcal{N}^{\mathcal{A}}\}$, and $R_{\text{lb}}(\mathcal{D}_{\mathcal{A}}; \mathbf{q})$ is the lower bound on the rate for given \mathbf{q} and $\mathcal{D}_{\mathcal{A}}$, given by

$$R_{\text{lb}}(\mathcal{D}_{\mathcal{A}}; \mathbf{q}) \triangleq \max_{\pi: \mathcal{I}_{|\mathcal{D}_{\mathcal{A}}|} \rightarrow \mathcal{D}_{\mathcal{A}}} \sum_{s=0}^{A-1} \sum_{i=1}^{\tilde{N}(\mathbf{d}_{\mathcal{A}})} \binom{A-i}{s} q_{\pi(i)}^s (1 - q_{\pi(i)})^{A-s} F \quad (21)$$

where $\mathcal{I}_{|\mathcal{D}_{\mathcal{A}}|} \triangleq \{1, \dots, |\mathcal{D}_{\mathcal{A}}|\}$ and $\pi: \mathcal{I}_{|\mathcal{D}_{\mathcal{A}}|} \rightarrow \mathcal{D}_{\mathcal{A}}$ is any bijective map from $\mathcal{I}_{|\mathcal{D}_{\mathcal{A}}|}$ to $\mathcal{D}_{\mathcal{A}}$.

Proof: See Appendix A. ■

P3 is a non-convex optimization problem, and the only difference between **P3** and **P0** are their objective functions $\bar{R}_{\text{MCCS}}(\mathbf{q})$ and $\bar{R}_{\text{lb}}(\mathbf{q})$.

Following the similar approach in Section IV-B1, we first formulate **P3** into an equivalent CGP problem. With the same auxiliary variables $x_n, n \in \mathcal{N}$, we add the same inequality constraints (9). Also, we introduce auxiliary variable $r_{\mathcal{D}_A, \pi}$ for $\pi: \mathcal{I}_{|\mathcal{D}_A|} \rightarrow \mathcal{D}_A, \mathcal{D}_A \subseteq \mathcal{N}$ and $\mathcal{A} \subseteq \mathcal{K}$. By (9), we replace the expression in (21) with $r_{\mathcal{D}_A, \pi}$ and add the following constraint

$$\sum_{s=0}^{A-1} \sum_{i=1}^{\tilde{N}(\mathbf{d}_A)} \binom{A-i}{s} (q_{\pi(i)})^s (x_{\pi(i)})^{A-s} F \leq r_{\mathcal{D}_A, \pi} \quad (22)$$

for given $\pi: \mathcal{I}_{|\mathcal{D}_A|} \rightarrow \mathcal{D}_A, \mathcal{D}_A \subseteq \mathcal{N}$ and $\mathcal{A} \subseteq \mathcal{K}$. Similar to the reformulation of **P0** to **P1**, with (22), we can reformulate **P3** into the following CGP.

$$\begin{aligned} \mathbf{P4}: \quad & \min_{\mathbf{q}, \mathbf{x}, \mathbf{r} \geq 0} E_{\mathcal{A}} \left[\sum_{\mathcal{D}_A \subseteq \mathcal{N}} \sum_{\mathbf{d}_A \in \mathcal{T}(\mathcal{D}_A)} \left(\prod_{k \in \mathcal{A}} p_{d_k} \right) r_{\mathcal{D}_A, \pi} \right] \\ \text{s.t.} \quad & (11), (12), (13) \text{ and} \\ & (r_{\mathcal{D}_A, \pi})^{-1} \sum_{s=0}^{A-1} \sum_{i=1}^{\tilde{N}(\mathbf{d}_A)} \binom{A-i}{s} (q_{\pi(i)})^s (x_{\pi(i)})^{A-s} F \leq 1, \\ & \mathcal{A} \subseteq \mathcal{K}, \mathcal{D}_A \subseteq \mathcal{N}, \forall \pi: \mathcal{I}_{|\mathcal{D}_A|} \rightarrow \mathcal{D}_A. \end{aligned} \quad (23)$$

Let $\bar{R}_{\text{lb}}^{\text{CGP}}(\mathbf{q}, \mathbf{x}, \mathbf{r})$ denote the objective function of **P4**. Following similar approach in Section IV-B1, in the i th iteration, for given $(\mathbf{q}^{(i)}, \mathbf{x}^{(i)})$, we formulate the following approximate optimization problem of **P4**.

$$\begin{aligned} \mathbf{P5}(\mathbf{q}^{(i)}, \mathbf{x}^{(i)}): \quad & (\mathbf{q}^{(i+1)}, \mathbf{x}^{(i+1)}, \mathbf{r}^{(i+1)}) = \underset{\mathbf{q}, \mathbf{x}, \mathbf{r} \geq 0}{\text{argmin}} \bar{R}_{\text{lb}}^{\text{CGP}}(\mathbf{q}, \mathbf{x}, \mathbf{r}) \\ \text{s.t.} \quad & (11), (12), (15), \text{ and } (23). \end{aligned}$$

Thus, we again use the successive GP approximation algorithm by iteratively solving **P5** $(\mathbf{q}^{(i)}, \mathbf{x}^{(i)})$ to obtain a stationary point of **P4**, which is summarized in Algorithm 4. Finally, by the equivalence of **P3** and **P4**, we obtain the stationary point of **P3** by Algorithm 4.

Algorithm 4 The Successive GP Approximation Algorithm for **P4**

Input: $K, M, N, \mathbf{p}, \mathbf{p}_a$

Output: $\bar{R}_{\text{lb}}^*, \mathbf{q}^*$

- 1: **Initialization:** Choose initial feasible point $(\mathbf{q}^{(0)}, \mathbf{x}^{(0)}, \mathbf{r}^{(0)})$, set $i = 0$.
 - 2: **repeat**
 - 3: Solve **P5** $(\mathbf{q}^{(i)}, \mathbf{x}^{(i)})$ to obtain $(\mathbf{q}^{(i+1)}, \mathbf{x}^{(i+1)}, \mathbf{r}^{(i+1)})$.
 - 4: Set $i = i + 1$.
 - 5: **until** $\bar{R}_{\text{lb}}^{\text{CGP}}(\mathbf{q}^{(i)}, \mathbf{x}^{(i)}, \mathbf{r}^{(i)})$ converges.
 - 6: $\bar{R}_{\text{lb}}^* = \bar{R}_{\text{lb}}^{\text{CGP}}(\mathbf{q}^{(i)}, \mathbf{x}^{(i)}, \mathbf{r}^{(i)}); \mathbf{q}^* = \mathbf{q}^{(i)}$.
-

B. Memory-Rate Tradeoff Characterization

We now compare the optimized D-MCCS in **P0** with the lower bound in **P3** and demonstrate the equivalence of the two problems in some specific cases. Since the difference between **P0** and **P3** is only in the average rate objective expression, it is sufficient to compare $\bar{R}_{\text{MCCS}}(\mathbf{q})$ and $\bar{R}_{\text{lb}}(\mathbf{q})$.

We first consider a special case where there are at most two users being active at the same time, *i.e.*, $A \leq 2$. Conditional

on $A \leq 2$, $\bar{R}_{\text{MCCS}}(\mathbf{q})$ in (8) and $\bar{R}_{\text{lb}}(\mathbf{q})$ in (20) are rewritten as

$$\bar{R}_{\text{MCCS}}(\mathbf{q}) = E_{\mathcal{A}} \left[\sum_{\mathbf{d}_A \in \mathcal{N}^A} \left(\prod_{k \in \mathcal{A}} p_{d_k} \right) R_{\text{MCCS}}(\mathbf{d}_A; \mathbf{q}) | A \leq 2 \right], \quad (24)$$

$$\bar{R}_{\text{lb}}(\mathbf{q}) = E_{\mathcal{A}} \left[\sum_{\mathcal{D}_A \subseteq \mathcal{N}} \sum_{\mathbf{d}_A \in \mathcal{T}(\mathcal{D}_A)} \left(\prod_{k \in \mathcal{A}} p_{d_k} \right) R_{\text{lb}}(\mathcal{D}_A; \mathbf{q}) | A \leq 2 \right]. \quad (25)$$

Comparing (24) and (25), we show in the following theorem that the lower bound in **P3** is tight.

Theorem 2. For the special case of no more than two active users at the same time, *i.e.*, $A \leq 2$, the average rate of the optimized D-MCCS in **P0** attains the lower bound in **P3**.

Proof: To show the equivalence of **P0** and **P3**, it is sufficient to show that $\bar{R}_{\text{MCCS}}(\mathbf{q})$ and $\bar{R}_{\text{lb}}(\mathbf{q})$ in (24) and (25) are equivalent. Comparing $\bar{R}_{\text{MCCS}}(\mathbf{q})$ and $\bar{R}_{\text{lb}}(\mathbf{q})$, we only need to examine $R_{\text{MCCS}}(\mathbf{d}_A; \mathbf{q})$ and $R_{\text{lb}}(\mathcal{D}_A; \mathbf{q})$ in (7) and (21). We compare $R_{\text{MCCS}}(\mathbf{d}_A; \mathbf{q})$ and $R_{\text{lb}}(\mathcal{D}_A; \mathbf{q})$ for the cases of $A = 1$ and $A = 2$ separately below.

Case 1: $A = 1$. Denote $\mathcal{A} = \{u_1\}$. In this case, $R_{\text{MCCS}}(\mathbf{d}_A; \mathbf{q})$ in (7) can be straightforwardly rewritten as

$$R_{\text{MCCS}}(\mathbf{d}_A; \mathbf{q}) = \sum_{S=\{u_1\}} \max_{k \in S} q_{d_k}^s (1 - q_{d_k})^{1-s} F = 1 - q_{d_{u_1}}.$$

For $\mathcal{D}_A = \{d_{u_1}\}$, we rewrite $R_{\text{lb}}(\mathcal{D}_A; \mathbf{q})$ in (21) as

$$R_{\text{lb}}(\mathcal{D}_A; \mathbf{q}) = 1 - q_{d_{u_1}} = R_{\text{MCCS}}(\mathbf{d}_A; \mathbf{q}), \quad |\mathcal{A}| = 1. \quad (26)$$

Case 2: $A = 2$. Denote $\mathcal{A} = \{u_1, u_2\}$. In this case, the two active users can either have the same or distinct file requests. We discuss the two cases in the following.

1) $d_{u_1} = d_{u_2}$: Two users request the same file, and we have $\tilde{N}(\mathbf{d}_A) = 1$. Without loss the generality, we denote leader group $\mathcal{U}_A = \{u_1\}$. By definition, the set of non-redundant groups is $\{\{u_1\}, \{u_1, u_2\}\}$. We can rewrite (7) as

$$\begin{aligned} R_{\text{MCCS}}(\mathbf{d}_A; \mathbf{q}) &= \sum_{S \in \{\{u_1\}, \{u_1, u_2\}\}} \max_{k \in S} q_{d_k}^s (1 - q_{d_k})^{2-s} F \\ &= (1 - q_{d_{u_1}})^2 + q_{d_{u_1}} (1 - q_{d_{u_1}}). \end{aligned}$$

Given the leader group $\mathcal{U}_A = \{u_1\}$, we have $\mathcal{D}_A = \{d_{u_1}\}$ and it is straightforward to rewrite (21) as

$$R_{\text{lb}}(\mathcal{D}_A; \mathbf{q}) = (1 - q_{d_{u_1}})^2 + q_{d_{u_1}} (1 - q_{d_{u_1}}) = R_{\text{MCCS}}(\mathbf{d}_A; \mathbf{q}). \quad (27)$$

2) $d_{u_1} \neq d_{u_2}$: When two users request different files and we have $N(\mathbf{d}_A) = 2$. The leader group is $\mathcal{U}_A = \{u_1, u_2\}$. Thus, we can rewrite $R_{\text{MCCS}}(\mathbf{d}_A; \mathbf{q})$ in (7) as

$$\begin{aligned} R_{\text{MCCS}}(\mathbf{d}_A; \mathbf{q}) &= \sum_{S \in \{\{u_1\}, \{u_2\}, \{u_1, u_2\}\}} \max_{k \in S} q_{d_k}^s (1 - q_{d_k})^{2-s} F \\ &= (1 - q_{d_{u_1}})^2 + (1 - q_{d_{u_2}})^2 \\ &\quad + \max\{q_{d_{u_1}} (1 - q_{d_{u_1}}), q_{d_{u_2}} (1 - q_{d_{u_2}})\}. \end{aligned}$$

Moreover, we also rewrite $R_{\text{lb}}(\mathcal{D}_A; \mathbf{q})$ in (21) as

$$R_{\text{lb}}(\mathcal{D}_A; \mathbf{q}) = \max\{(1 - q_{d_{u_1}})^2 + (1 - q_{d_{u_2}})^2 + q_{d_{u_1}} (1 - q_{d_{u_1}}),$$

$$\begin{aligned} & (1 - q_{d_{u_1}})^2 + (1 - q_{d_{u_2}})^2 + q_{d_{u_2}}(1 - q_{d_{u_2}}) \} \\ & = R_{\text{MCCS}}(\mathbf{d}_A; \mathbf{q}). \end{aligned} \quad (28)$$

From (27) and (28), we conclude $\bar{R}_{\text{MCCS}}(\mathbf{q}) = \bar{R}_{\text{lb}}(\mathbf{q})$ for $A = 2$. Combining the result in (26), we prove that $\bar{R}_{\text{MCCS}}(\mathbf{q}) = \bar{R}_{\text{lb}}(\mathbf{q})$ for $A = 1, 2$. ■

Theorem 2 shows that if there are no more than two active users at the same time, then the optimized D-MCCS is an optimal decentralized caching scheme. In this case, the optimized D-MCCS characterizes the exact memory-rate tradeoff for decentralized caching under nonuniform file popularity. Moreover, the result also implies that zero-padding used in the delivery phased of D-MCCS incurs no loss of optimality in this case.

In the general scenario where the active user set is not limited to two users, although $\bar{R}_{\text{MCCS}}(\mathbf{q})$ and $\bar{R}_{\text{lb}}(\mathbf{q})$ may not be the same, the optimal placement solution \mathbf{q}^* to **P0**, **P3** may still be the same for certain system configuration $(N, K, M, \mathbf{p}, \mathbf{p}_a)$. The following proposition describes the result in this case.

Proposition 2. If \mathbf{q}^* with $q_1^* = \dots = q_N^*$ is the optimal solution to both **P0** and **P3**, then $\bar{R}_{\text{MCCS}}(\mathbf{q}^*) = \bar{R}_{\text{lb}}(\mathbf{q}^*)$. *i.e.*, the lower bound in **P3** is attained by the optimized D-MCCS in **P0**.

Proof: For $q_1^* = \dots = q_N^*$, based on (5), the length of coded message \mathcal{C}_S corresponding to $\mathcal{S} \in \mathcal{Q}^{s+1}$ is given by

$$\max_{k \in \mathcal{S}} (q_{d_k}^*)^s (1 - q_{d_k}^*)^{A-s} F = (q_1^*)^s (1 - q_1^*)^{A-s} F.$$

Following this, based on (7), we have

$$R_{\text{MCCS}}(\mathbf{d}_A; \mathbf{q}^*) = \sum_{s=0}^{A-1} \sum_{\mathcal{S} \in \mathcal{Q}^{s+1}} (q_1^*)^s (1 - q_1^*)^{A-s} F. \quad (29)$$

The two summations in (29) represents the number of all the non-redundant groups. By the definition, the non-redundant groups are the active user subset include at least one users in the leader group \mathcal{U}_A . Denote $\mathcal{U}_A \triangleq \{u_1, \dots, u_{\tilde{N}(\mathbf{d}_A)}\}$. Among all user subsets in \mathcal{Q}^{s+1} , there are $\binom{A-i}{s}$ subsets including user $\{u_1, \dots, u_i\}$. By considering $i = 1, \dots, \tilde{N}(\mathbf{d}_A)$, we can rewrite (29) as

$$\begin{aligned} R_{\text{MCCS}}(\mathbf{d}_A; \mathbf{q}^*) &= \sum_{s=0}^{A-1} \sum_{i=1}^{\tilde{N}(\mathbf{d}_A)} \binom{A-i}{s} (q_1^*)^s (1 - q_1^*)^{A-s} F \\ &= R_{\text{lb}}(\mathcal{D}_A; \mathbf{q}^*). \end{aligned} \quad (30)$$

Thus, we can conclude that $\bar{R}_{\text{MCCS}}(\mathbf{q}^*) = \bar{R}_{\text{lb}}(\mathbf{q}^*)$. ■

Proposition 2 indicates that if the optimal placement \mathbf{q}^* is symmetric for all files, $q_1^* = \dots = q_N^*$, the optimized D-MCCS is an optimal decentralized caching scheme that characterizes the exact memory-rate tradeoff. One known example is the special case of uniform file popularity. In this case, the optimized D-MCCS (**P0**) and the lower bound (**P3**) have the same optimal solution \mathbf{q}^* with q_n^* 's being all identical, and the result $\bar{R}_{\text{MCCS}}(\mathbf{q}^*) = \bar{R}_{\text{lb}}(\mathbf{q}^*)$ has been shown in [18].

Finally, we point out that our numerical study in Section VI shows that the gap between the optimized D-MCCS and the lower bound in **P3** is very small in general. This indicates

that the performance of the optimized D-MCCS is very close to the optimal decentralized caching.

VI. SIMULATIONS

We now provide our numerical study on the performance of the optimized D-MCCS in **P0** and the proposed lower bound for decentralized caching in **P3**. We set $N = 6$ files and $K = 4$ users. We generate the nonuniform file popularities using the Zipf distribution with $p_n = n^{-\theta} / \sum_{i=1}^N i^{-\theta}$, where θ is the Zipf parameter. We set the probability of each user being active as $p_{a,k} = 0.5, k \in \mathcal{K}$. We use \bar{R} to generally indicate the average rate obtained by various schemes considered. We consider our proposed two schemes (Algorithms 2 and 3) for solving **P0** to optimize D-MCCS, Algorithm 4 for the lower bound in **P3**. We set the convergence criterion for both Algorithms 2 and 4 as the difference in the average rate over consecutive iterations is less than $1e-4$. For comparison, we also consider the existing well-known decentralized scheme based on D-CCS, including the optimized D-CCS [17] and the file-grouping-based schemes in [15] and [16].

In Fig. 2, we plot \bar{R} vs. cache size M under different schemes for $\theta = 0.56$ (the same as [5]). For the optimized D-MCCS in **P0**, we observe that the average rate \bar{R} achieved by Algorithms 2 and 3 are nearly identical. This indicates that the low-complexity two-file-group-based solution is close to the stationary points obtained by the successive GP approximation algorithm. Among the caching schemes compared, the optimized D-MCCS provides the lowest average rate for all values of M . The gap between the optimized D-MCCS and the optimized D-CCS reduces as M increases. This is mainly because there are more redundant groups in the coded delivery phase for a smaller value of M , and, as a result, the D-MCCS improved upon the D-CCS more. The average rates obtained by Algorithms 2 and 3 are both very close to the lower bound in **P3**, with only a very small gap observed for $M \in [1, 3]$. This shows that the performance of the optimized D-MCCS is close to that under the optimal decentralized caching.

In Fig. 3, we consider a larger value of $\theta = 1.2$ for a more skewed file popularity distribution to study \bar{R} vs. M . The average rates obtained by Algorithms 2 and 3 are again nearly identical. However, it is interesting to see that for $M = 1$ and 2, Algorithm 3 achieves a lower average rate than Algorithm 2 does. This shows that in some cases, the two-file-group-based solution could perform even better than the successive GP approximation algorithm, which has a much higher computational complexity. Among all the schemes compared, the optimized D-MCCS again achieves the lowest average rate for all values of M . The gap between optimized D-MCCS and the lower bound is again very small in general. This demonstrates the optimized D-MCCS is close to optimal for decentralized caching.

VII. CONCLUSION

In this paper, we studied the memory-rate tradeoff for decentralized caching with nonuniform file popularity. Focusing on the D-MCCS, we formulated the cache placement optimization problem and developed two algorithms to solve

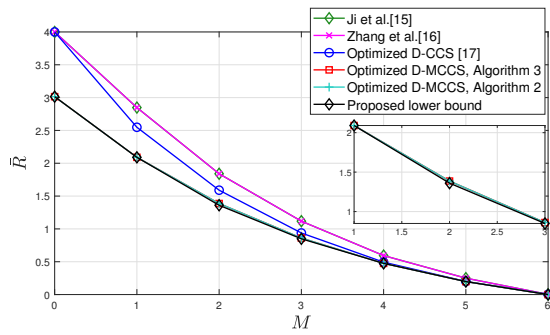


Fig. 2. Average rate \bar{R} vs. cache size M ($N = 6$, $K = 4$, Zipf file popularity distribution with $\theta = 0.56$).

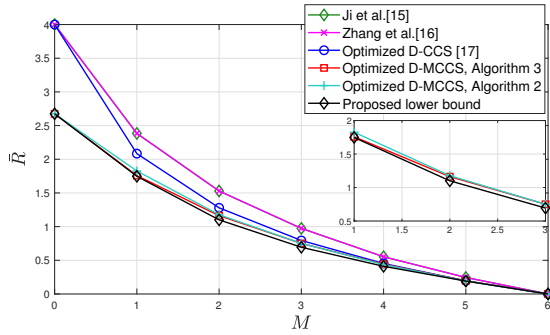


Fig. 3. Average rate \bar{R} vs. cache size M ($N = 6$, $K = 4$, Zipf file popularity distribution with $\theta = 1.2$).

this non-convex problem: a successive GP approximation algorithm to compute a stationary point of the optimization problem, and a low-complexity simple two-file-group-based approximate algorithm. We further proposed a lower bound through a non-convex optimization problem, for which we developed an algorithm to achieve a stationary point. For the special case of no more than two active users at a time, we show that the optimized D-MCCS attains the lower bound and thus characterizes the exact memory-rate tradeoff. For general cases, a condition for the D-MCCS attaining the lower bound is also identified. Our numerical study showed that the optimized D-MCCS in general achieves an average rate that is very close to the lower bound.

APPENDIX A PROOF OF THEOREM 1

Proof: The proof follows the genie-based approach used in developing the lower bound for the centralized uncoded cache placement under uniform or nonuniform file popularity [7], [18] or nonuniform cache sizes [9]. For a given file request vector $\mathcal{D}_{\mathcal{A}}$ by the active users in \mathcal{A} , the average delivery rate must satisfy [7]

$$R(\mathcal{D}_{\mathcal{A}}; \mathbf{q}) \geq \max_{\pi: \mathcal{I}_{|\mathcal{D}_{\mathcal{A}}|} \rightarrow \mathcal{D}_{\mathcal{A}}} \sum_{i=1}^{\tilde{N}(\mathbf{d}_{\mathcal{A}})} \sum_{s=1}^{A-1} \binom{A-s}{i} a_{\pi(i), s} \quad (31)$$

where $a_{\pi(i), s}$ is the number of bits of file $\pi(i)$ cached exclusively by any user subset $\mathcal{S} \in \mathcal{A}$ with $|\mathcal{S}| = s$ users. With decentralized placement, as shown in (3), the number of bits cached by any s active users in \mathcal{A} is $a_{\pi(i), s} = q_{\pi(i)}^s (1 - q_{\pi(i)})^{A-s} F$. Following this, under decentralized placement, we can rewrite (31) as

$$R(\mathcal{D}_{\mathcal{A}}; \mathbf{q}) \geq \max_{\pi: \mathcal{I}_{|\mathcal{D}_{\mathcal{A}}|} \rightarrow \mathcal{D}_{\mathcal{A}}} \sum_{i=1}^{\tilde{N}(\mathbf{d}_{\mathcal{A}})} \sum_{s=1}^{A-1} \binom{A-s}{i} q_{\pi(i)}^s (1 - q_{\pi(i)})^{A-s} F.$$

which is the lower bound on the delivery rate for a given $\mathcal{D}_{\mathcal{A}}$ and \mathbf{q} defined in (21). By averaging $R_{\text{lb}}(\mathcal{D}_{\mathcal{A}}; \mathbf{q})$ over all possible $\mathcal{D}_{\mathcal{A}} \subseteq \mathcal{N}$ and \mathcal{A} over all possible $\mathcal{A} \subseteq \mathcal{K}$, we obtain the general lower bound $\bar{R}_{\text{lb}}(\mathbf{q})$ the average rate w.r.t \mathbf{q} in (20). The final lower bound on average rate is obtained by optimizing \mathbf{q} to minimize $\bar{R}_{\text{lb}}(\mathbf{q})$, which is the optimization problem **P4** as presented in Theorem 1. ■

REFERENCES

- [1] G. S. Paschos, G. Iosifidis, M. Tao, D. Towsley, and G. Caire, "The role of caching in future communication systems and networks," *IEEE J. Sel. Areas Commun.*, vol. 36, pp. 1111–1125, Sep. 2018.
- [2] M. A. Maddah-Ali and U. Niesen, "Fundamental limits of caching," *IEEE Trans. Inform. Theory*, vol. 60, pp. 2856–2867, Mar. 2014.
- [3] Y. Deng and M. Dong, "Subpacketization level in optimal placement for coded caching with nonuniform file popularities," in *the 53rd Asilomar Conf. on Signals, Systems, and Computers*, Nov. 2019, pp. 1294–1298.
- [4] Y. Deng and M. Dong, "Fundamental structure of optimal cache placement for coded caching with nonuniform demands," *arXiv preprint arXiv:1912.01082*, Apr. 2020.
- [5] A. M. Daniel and W. Yu, "Optimization of heterogeneous coded caching," *IEEE Trans. Inform. Theory*, vol. 66, pp. 1893–1919, Mar. 2020.
- [6] S. Jin, Y. Cui, H. Liu, and G. Caire, "Uncoded placement optimization for coded delivery," *arXiv preprint arXiv:1709.06462*, Jul. 2018.
- [7] Y. Deng and M. Dong, "Memory-rate tradeoff for caching with uncoded placement under nonuniform file popularity," in *the 54th Asilomar Conf. on Signals, Systems, and Computers*, Nov. 2020, pp. 336–340.
- [8] M. Ji, G. Caire, and A. F. Molisch, "Fundamental limits of caching in wireless D2D networks," *IEEE Trans. Inform. Theory*, vol. 62, pp. 849–869, Feb. 2016.
- [9] A. M. Ibrahim, A. A. Zewail, and A. Yener, "Coded caching for heterogeneous systems: An optimization perspective," *IEEE Trans. Commun.*, vol. 67, pp. 5321–5335, Aug. 2019.
- [10] S. A. Saberali, L. Lampe, and I. F. Blake, "Full characterization of optimal uncoded placement for the structured clique cover delivery of nonuniform demands," *IEEE Trans. Inform. Theory*, vol. 66, pp. 633–648, Jan. 2020.
- [11] M. A. Maddah-Ali and U. Niesen, "Decentralized coded caching attains order-optimal memory-rate tradeoff," *IEEE/ACM Trans. Netw.*, vol. 23, pp. 1029–1040, Aug. 2015.
- [12] M. Mohammadi Amiri, Q. Yang, and D. Gndz, "Decentralized caching and coded delivery with distinct cache capacities," *IEEE Trans. Commun.*, vol. 65, pp. 4657–4669, Nov. 2017.
- [13] L. Zheng, Q. Chen, Q. Yan, and X. Tang, "Decentralized coded caching scheme with heterogeneous file sizes," *IEEE Trans. Veh. Technol.*, vol. 69, pp. 818–827, Jan. 2020.
- [14] U. Niesen and M. A. Maddah-Ali, "Coded caching with nonuniform demands," *IEEE Trans. Inform. Theory*, vol. 63, pp. 1146–1158, Dec. 2017.
- [15] M. Ji, A. M. Tulino, J. Llorca, and G. Caire, "Order-optimal rate of caching and coded multicasting with random demands," *IEEE Trans. Inform. Theory*, vol. 63, pp. 3923–3949, Apr. 2017.
- [16] J. Zhang, X. Lin, and X. Wang, "Coded caching under arbitrary popularity distributions," *IEEE Trans. Inform. Theory*, vol. 64, pp. 349–366, Nov. 2018.
- [17] Q. Wang, Y. Cui, S. Jin, J. Zou, C. Li, and H. Xiong, "Optimization-based decentralized coded caching for files and caches with arbitrary sizes," *IEEE Trans. Commun.*, vol. 68, pp. 2090–2105, Apr. 2020.
- [18] Q. Yu, M. A. Maddah-Ali, and A. S. Avestimehr, "The exact rate-memory tradeoff for caching with uncoded prefetching," *IEEE Trans. Inform. Theory*, vol. 64, pp. 1281–1296, Feb. 2018.
- [19] M. Avriel, *Advances in Geometric Programming*. New York: Plenum-Press, 1980.
- [20] M. Chiang, C. W. Tan, D. P. Palomar, D. O'neill, and D. Julian, "Power control by geometric programming," *IEEE Trans. Wireless Commun.*, vol. 6, pp. 2640–2651, Jul. 2007.