Uncoded Placement Optimization for Coded Delivery

Sian Jin, Ying Cui, Hui Liu
Shanghai Jiao Tong University, China

Giuseppe Caire
Technical University of Berlin, Germany

Abstract— **Existing coded caching schemes fail to simultaneously achieve efficient content placement for non-uniform file popularity and efficient content delivery in the presence of common requests, and hence may not achieve desirable average load under a non-uniform, possibly very skewed, popularity distribution. In addition, existing coded caching schemes usually require the splitting of a file into a large number of subfiles, i.e., high subpacketization, and hence may cause huge implementation complexity. To address the above two challenges, we first present a class of centralized coded caching schemes consisting of a general content placement strategy specified by a file partition parameter, enabling efficient and flexible content placement, and a specific content delivery strategy, enabling load reduction by exploiting common requests of different users. Then we consider two cases, namely, the case without considering the subpacketization issue and the case considering the subpacketization issue. In the first case, we formulate the coded caching optimization problem over the considered class of schemes with $N2^K$ variables to minimize the average load under an arbitrary file popularity. Imposing some conditions on the file partition parameter, we transform the original optimization problem into a linear optimization problem with $N(K+1)$ variables under an arbitrary file popularity and a linear optimization problem with $K+1$ variables under the uniform file popularity. We also show that Yu *et al.*'s centralized coded caching scheme corresponds to an optimal solution of our problem. In the second case, taking into account the subpacketization issue, we first formulate the coded caching optimization problem over the considered class of schemes to minimize the average load under an arbitrary file popularity subject to a subpacketization constraint involving the ℓ_0-norm. By imposing the same conditions and using an exact DC (difference of two convex functions) reformulation method, we convert the original problem with $N2^K$ variables into a simplified DC problem with $N(K+1)$ variables. Then, we use a DC algorithm to solve the simplified DC problem.**

I. Introduction

Recently, a new class of caching schemes for content placement in user caches, referred to as *coded caching* [1], have received significant interest. In [1], Maddah-Ali and Niesen consider a system with one server connected through a shared error-free link to K users. The server has a library of N files, and each user has an isolated cache memory of M files. They formulate a caching problem, consisting of two phases, i.e., uncoded content placement and coded content delivery. Each user can obtain the requested file based on the received coded-multicast messages and the contents stored in its cache. The goal of [1] is to reduce the worst-case (over all possible requests) load of the shared link in the delivery phase. In [2], Jin *et al.* consider a different class of centralized coded caching schemes specified by a general file partition parameter, and optimize the parameter to minimize the average load within the class under an arbitrary file popularity. In [3], the parameter-based coded caching design approach in [2] is generalized to minimize the average load in a heterogeneous setting with nonuniform file popularity, cache size and file size. In [4], Yu *et al.* propose a centralized coded caching scheme where the delivery strategy exploits the chance of load reduction in common requests of different users. The scheme devised by Yu *et al.* is proved to be optimal for the worst-case load and average load under the uniform popularity.

Note that the delivery strategies in [1]–[3] do not capture the opportunity of load reduction in common requests of different users, and the placement strategies in [1] and [4] allocate the same fraction of memory to each file without reflecting popularity difference. Therefore, the coded caching strategies in [1]–[4] may not achieve desirable average load under a non-uniform, possibly very skewed, popularity distribution. It is not known how to simultaneously achieve efficient content placement for non-uniform file popularity and efficient content delivery in the presence of common requests.

Another limitation of previous works is the issue of high subpacketization, i.e., the number of non-overlapping subfiles for each file is large. In [5]–[8], the authors tackle the subpacketization issue for decentralized coded caching. Specifically, in [5], a user grouping method is proposed to lower the subpacketization at the expense of larger worst-case load. In [6], based on a novel user grouping idea, the authors propose a new order-optimal decentralized random coded caching scheme, which induces lower subpacketization than that in [5] at the same worst-case load. To reduce the average loads in practical regimes of finite subpacketization, the authors in [7] and [8] propose decentralized coded caching schemes based on hierarchy greedy local graph coloring. In [9]–[11], the authors tackle the subpacketization issue for centralized coded caching. Specifically, in [9], Tang *et al.* connect coded caching to resolvable combinatorial designs and propose a centralized coded caching scheme where the subpacketization is exponential with respect to (w.r.t.) the number of users but significantly lower than the centralized coded caching scheme of [1] at the cost of a marginal increase in the worst-case load. In [10], Shanmugam *et al.*

The work of Y. Cui was supported by National Science Foundation of China under Grant 61401272 and Grant 61521062, Science and Technology on Communication Networks Laboratory Foundation, as well as Shanghai Key Laboratory Funding STCSM15DZ2270400.

connect coded caching to Ruzsa-Szeméredi graphs and show the existence of a centralized coded caching scheme where the subpacketization grows linearly with the number of users when the number of users is very large. In [11], the authors propose a centralized coded caching scheme with low subpacketization based on Pareto-optimal PDA (placement delivery array). Note that the coded caching schemes in [5], [9]–[11] addressing the subpacketization issue are applicable only for certain system parameters (e.g., the number of users, the number of files, cache size, etc.). Furthermore, the coded caching schemes in [5], [6], [9]–[11] are not based on optimizations and cannot guarantee desirable average loads under subpacketization constraints.

In this paper, we would like to address the above challenges in the same centralized setting as in [1]–[4], [9]–[11], with the focus on minimizing the average load under an arbitrary file popularity in two cases, namely, the case without considering the subpacketization issue and the case considering the subpacketization issue. We present a class of coded caching schemes consisting of a general content placement strategy specified by a file partition parameter, enabling efficient and flexible content placement, and a specific content delivery strategy, enabling load reduction by exploiting common requests of different users. Then, we focus on the average load minimization irrespectively of the subpacketization issue. In this case, we formulate the coded caching optimization problem over the considered class of schemes with $N2^K$ variables to minimize the average load under an arbitrary file popularity. The average load expression is not tractable due to the complex delivery strategy. Therefore, we impose some conditions on the parameter to simplify the average load expression under an arbitrary file popularity and the uniform file popularity respectively, by connecting the file request event to the "balls into bins" problem. Based on the simplified expressions, we transform the original optimization problem into a linear optimization problem with $N(K+1)$ variables under an arbitrary file popularity and a linear optimization problem with $K+1$ variables under the uniform file popularity, which are much easier to solve than the original problem. We also show that Yu *et al.*'s centralized coded caching scheme corresponds to an optimal solution of our problem and the imposed conditions are optimal properties for the uniform file popularity. Next, we consider the average load minimization subject to a constraint on subpacketization. In this case, we first formulate the coded caching optimization problem over the considered class of schemes to minimize the average load under an arbitrary file popularity subject to a subpacketization constraint in terms of the ℓ_0-norm of the file partition parameter. To the best of our knowledge, this is the first work explicitly considering a subpacketization constraint in optimization based coded caching design. By imposing the same conditions as in the case without considering the subpacketization issue and using the exact DC (difference of two convex functions) reformulation method in [12], we convert the original problem with $N2^K$ variables into a simplified DC problem with $N(K+1)$ variables. Then, we use a DC algorithm to solve the simplified DC problem. Numerical results reveal that the imposed conditions do not affect the optimality of the original problem under an arbitrary file popularity in the case without considering the subpacketization constraint. Furthermore, our results demonstrate that the optimized coded caching scheme without considering the subpacketization constraint outperforms those in [1], [2], [4] in terms of the average load, and the optimized coded caching scheme considering the subpacketization constraint outperforms those in [9] and [11] in terms of both the average load and application region.

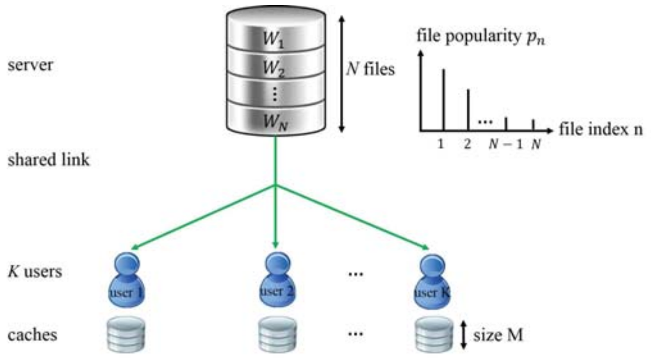


Fig. 1: Problem setup for coded caching.

II. Centralized Coded Caching

A. Problem Setting

As in [1]–[4], [9]–[11], we consider a system with one server connected through a shared error-free link to $K \in \mathbb{N}_{>0}$ users (see Fig. 1), where $\mathbb{N}_{>0}$ denotes the set of all positive integers. The server has access to a library of $N \in \mathbb{N}_{>0}$ files, denoted by $W_1, \ldots, W_N$, each consisting of $F \in \mathbb{N}_{>0}$ indivisible data units. Let $\mathcal{N} \triangleq \{1, 2, \ldots, N\}$ and $\mathcal{K} \triangleq \{1, 2, \ldots K\}$ denote the set of file indices and the set of user indices, respectively. Each user has an isolated cache memory of MF data units, for some real number $M \in [0, N]$. Let Z_k denote the cache content for user k. The system operates in two phases, i.e., a placement phase and a delivery phase [1]. In the placement phase, each user is able to fill the content of its cache using the library of N files. In the delivery phase, each user randomly and independently requests one file in $\mathcal{N}$ according to file popularity distribution $\mathbf{p} \triangleq (p_n)_{n=1}^N$, where p_n denotes the probability of a user requesting file W_n and $\sum_{n=1}^N p_n = 1$. Without loss of generality, we assume $p_1 \geq p_2 \geq \ldots \geq p_N$. Let $D_k \in \mathcal{N}$ denote the index of the file requested by user $k \in \mathcal{K}$, and let $\mathbf{D} \triangleq (D_1, \cdots, D_K) \in \mathcal{N}^K$ denote the requests of all the K users. The server replies to these K requests by sending messages over the shared link, which are observed by all the K users. Each user should be able to recover its requested file from the messages received over the shared link and its cache content. Our goal is to minimize the average load of the shared link under an arbitrary file popularity.

B. Centralized Coded Caching Scheme

In this part, we present a class of centralized coded caching schemes utilizing a general uncoded placement strategy and a specific coded delivery strategy, which are specified by a general file partition parameter, as summarized in Alg. 1. The general uncoded placement strategy is the same as that in [2]. We present it here for completeness. For all $n \in \mathcal{N}$, file W_n is partitioned into 2^K nonoverlapping subfiles $W_{n,\mathcal{S}}$, $\mathcal{S} \subseteq \mathcal{K}$, i.e., $W_n = (W_{n,\mathcal{S}} : \mathcal{S} \subseteq \mathcal{K})$. If the number of data units in a subfile is zero, then there is no need to consider this subfile. Thus, 2^K is the maximum number of non-overlapping subfiles of a file. We say subfile $W_{n,\mathcal{S}}$ is of type s if $|\mathcal{S}| = s$ [2]. User k stores $W_{n,\mathcal{S}}$, $n \in \mathcal{N}, k \in \mathcal{S}, \mathcal{S} \subseteq \mathcal{K}$ in its cache, i.e., $Z_k = (W_{n,\mathcal{S}} : n \in \mathcal{N}, k \in \mathcal{S}, \mathcal{S} \subseteq \mathcal{K})$. Let $x_{n,\mathcal{S}}$ denote the ratio between the number of data units in $W_{n,\mathcal{S}}$ and the number of data units in W_n (i.e., F). Let $\mathbf{x}_n \triangleq (x_{n,\mathcal{S}})_{\mathcal{S} \subseteq \mathcal{K}}$. Let $\mathbf{x} \triangleq (\mathbf{x}_n)_{n \in \mathcal{N}}$ denote the file partition parameter (vector), which will be optimized to minimize the average load in Section III. Thus, $\mathbf{x}$ satisfies

$$0 \le x_{n,\mathcal{S}} \le 1, \quad \forall \mathcal{S} \subseteq \mathcal{K},\ n \in \mathcal{N}, \tag{1}$$

$$\sum_{s=0}^{K} \sum_{\mathcal{S} \in \{\widehat{\mathcal{S}} \subseteq \mathcal{K} : |\widehat{\mathcal{S}}| = s\}} x_{n,\mathcal{S}} = 1, \quad \forall n \in \mathcal{N}, \tag{2}$$

$$\sum_{n=1}^{N} \sum_{s=1}^{K} \sum_{\mathcal{S} \in \{\widehat{\mathcal{S}} \subseteq \mathcal{K} : |\widehat{\mathcal{S}}| = s, k \in \widehat{\mathcal{S}}\}} x_{n,\mathcal{S}} \le M, \quad \forall k \in \mathcal{K}, \tag{3}$$

where (1) and (2) represent the file partition constraints and (3) represents the cache memory constraint.

The coded delivery strategy is an extension of that in [4]. For all $\mathbf{D} \in \mathcal{N}^K$, let $\underline{\mathcal{D}}(\mathbf{D}) \triangleq \{D_k : k \in \mathcal{K}\}$ denote the set of distinct files in $\mathbf{D}$. For all $n \in \underline{\mathcal{D}}(\mathbf{D})$, the server arbitrarily selects user $k_n \in \mathcal{K}$ such that $D_{k_n} = n$. Let $\underline{\mathcal{K}}(\mathbf{D}) \triangleq \{k_n : n \in \underline{\mathcal{D}}(\mathbf{D})\}$ denote the set of representative users that request $|\underline{\mathcal{D}}(\mathbf{D})|$ different files. Each user $k \in \mathcal{S}$ requests subfile $W_{D_k,\mathcal{S}\setminus\{k\}}$, for all subset $\mathcal{S} \subseteq \mathcal{K}$. The server broadcasts coded-multicast message $\oplus_{k \in \mathcal{S}} W_{D_k,\mathcal{S}\setminus\{k\}}$ for all subset $\mathcal{S} \subseteq \mathcal{K}$ that satisfies $\mathcal{S} \cap \underline{\mathcal{K}}(\mathbf{D}) \neq \emptyset$, and all subfiles in the coded-multicast message are zero-padded to the length of the longest subfile.[1] By Lemma 1 of [4], we can conclude that each user can decode the requested file based on the received coded-multicast messages and the contents stored in its cache.

Remark 1: The uncoded placement in this paper is more general than that in [1], [4], [9]–[11], and it can be optimized to minimize the average load under an arbitrary file popularity (see Section III and Section IV). The coded delivery in this paper is more efficient than that in [1]–[3], [5], [6], [9]–[11], since it avoids transmitting the redundant coded-multicast messages $\oplus_{k \in \mathcal{S}} W_{D_k,\mathcal{S}\setminus\{k\}}$, $\mathcal{S} \subseteq \mathcal{K} \setminus \underline{\mathcal{K}}(\mathbf{D})$ in the presence of common requests [4].

C. Average Load

Let $R_{\text{avg}}(K, N, M, \mathbf{x})$ denote the average load for serving the K users with cache size M under a given file partition parameter $\mathbf{x}$, where the average is taken over random requests $\mathbf{D}$ for N files, according to an arbitrary file popularity distribution $\mathbf{p}$. By Alg. 1, we have

$$R_{\text{avg}}(K, N, M, \mathbf{x}) = \sum_{\mathbf{d} \in \mathcal{N}^K} \left(\prod_{k=1}^{K} p_{d_k} \right) \sum_{\mathcal{S} \in \{\widehat{\mathcal{S}} \subseteq \mathcal{K} : \widehat{\mathcal{S}} \cap \underline{\mathcal{K}}(\mathbf{D}) \neq \emptyset\}} \max_{k \in \mathcal{S}} x_{d_k,\mathcal{S}\setminus\{k\}}, \tag{4}$$

where $\mathbf{d} \triangleq (d_1, \ldots, d_K) \in \mathcal{N}^K$ and $\max_{k \in \mathcal{S}} x_{d_k,\mathcal{S}\setminus\{k\}}$ is the length of the coded message $\oplus_{k \in \mathcal{S}} W_{d_k,\mathcal{S}\setminus\{k\}}$ divided by F.

From (4), we can observe that the file partition parameter $\mathbf{x}$ fundamentally affects the average load $R_{\text{avg}}(K, N, M, \mathbf{x})$. In Section III, we would like to find an optimal file partition parameter to minimize the average load in (4). The optimal file partition parameter may correspond to high subpacketization, and hence a high file partition cost. To avoid high subpacketization, in Section IV, we would like to find an optimal file partition parameter to minimize the average load in (4) under a subpacketization constraint.

Algorithm 1 Parameter-based Centralized Coded Caching

```
placement procedure
1: for all $k \in \mathcal{K}$ do
2:     $Z_k \leftarrow (W_{n,\mathcal{S}} : n \in \mathcal{N}, k \in \mathcal{S}, \mathcal{S} \subseteq \mathcal{K})$
3: end for
delivery procedure
1: for $s = K, K-1, \cdots, 1$ do
2:     for $\mathcal{S} \subseteq \mathcal{K} : |\mathcal{S}| = s, \mathcal{S} \cap \underline{\mathcal{K}}(\mathbf{D}) \neq \emptyset$ do
3:         server sends $\oplus_{k \in \mathcal{S}} W_{D_k,\mathcal{S}\setminus\{k\}}$
4:     end for
5: end for
```

III. Average Load Minimization without Subpacketization Constraint

In this section, we minimize the average load by optimizing the file partition parameter without considering any subpacketization constraint.

A. Problem Formulation

We would like to minimize the average load under the file partition constraints in (1) and (2) as well as the cache memory constraint in (3).

Problem 1 (Optimization for Arbitrary File Popularity):

$$R^*_{\text{avg}}(K, N, M) \triangleq \min_{\mathbf{x}} \quad R_{\text{avg}}(K, N, M, \mathbf{x}) \qquad s.t. \quad (1), (2), (3),$$

where $R_{\text{avg}}(K, N, M, \mathbf{x})$ is given by (4).

The objective function of Problem 1 is convex, as it is a positive weighted sum of convex piecewise linear functions. In addition, the constraints of Problem 1 are linear. Hence, Problem 1 is a convex optimization problem. The number of variables in Problem 1 is $N2^K$. Thus, the complexity of

[1]Note that in [4], since all files are partitioned into subfiles of type $t \in \{0, 1, \ldots, K\}$, only coded-multicast messages $\oplus_{k \in \mathcal{S}} W_{D_k,\mathcal{S}\setminus\{k\}}$ satisfying $\mathcal{S} \subseteq \mathcal{K}$, $\mathcal{S} \cap \underline{\mathcal{K}}(\mathbf{D}) \neq \emptyset$ and $|\mathcal{S}| = t + 1$ are transimitted.

Problem 1 is huge, especially when K and N are large. In Section III-B and Section III-C, we shall focus on deriving simplified formulations for Problem 1 to facilitate low-complexity optimal solutions under an arbitrary popularity distribution and the uniform popularity distribution, respectively.

B. Optimization for Arbitrary Popularity without Subpacketization Constraint

First, we present two structural conditions on the file partition parameter $\mathbf{x}$ to facilitate a low-complexity solution of Problem 1.

Condition 1 (Symmetry w.r.t. Type): For all $n \in \mathcal{N}$ and $s \in \{0, 1, \cdots, K\}$, the values of $x_{n,\mathcal{S}}, \mathcal{S} \in \{\widehat{\mathcal{S}} \subseteq \mathcal{K} : |\widehat{\mathcal{S}}| = s\}$ are the same.

Condition 1 indicates that, for all $n \in \mathcal{N}$ and $s \in \{0, 1, \cdots, K\}$, subfiles $W_{n,\mathcal{S}}$, $\mathcal{S} \in \{\widehat{\mathcal{S}} \subseteq \mathcal{K} : |\widehat{\mathcal{S}}| = s\}$ have the same size. Recall that all subfiles in one coded-multicast message are zero-padded to the length of the longest subfile in the coded-multicast message, causing the "bit waste" effect [2]. Thus, imposing Condition 1 can reduce the variance of the lengths of messages involved in the coded-multicast XOR operations, hence addressing "bit waste" problem and increasing coded-multicasting opportunities. By Condition 1, we can set

$$x_{n,\mathcal{S}} = y_{n,s}, \quad \forall \mathcal{S} \subseteq \mathcal{K},\ n \in \mathcal{N}, \tag{5}$$

where $s = |\mathcal{S}| \in \{0, 1, \cdots, K\}$. Here, $y_{n,s}$ can be viewed as the ratio between the number of data units in each subfile of type s of file W_n and the number of data units in file W_n (i.e., F). Let $\mathbf{y}_n \triangleq (y_{n,s})_{s \in \{0,1,\cdots,K\}}$ and $\mathbf{y} \triangleq (\mathbf{y}_n)_{n \in \mathcal{N}}$.

Condition 2 (Monotonicity w.r.t. Popularity): For all $n \in \{1, 2, \ldots, N-1\}$ and $s \in \{1, 2, \cdots, K\}$, when $p_n \geq p_{n+1}$,

$$y_{n,s} \geq y_{n+1,s}. \tag{6}$$

Condition 2 indicates that, for all $n \in \{1, 2, \ldots, N-1\}$ and $s \in \{1, 2, \cdots, K\}$, when $p_n \geq p_{n+1}$, the size of subfiles $W_{n,\mathcal{S}}$, $\mathcal{S} \subseteq \{\widehat{\mathcal{S}} \subseteq \mathcal{K} : |\widehat{\mathcal{S}}| = s\}$ is no smaller than that of subfiles $W_{n+1,\mathcal{S}}$, $\mathcal{S} \subseteq \{\widehat{\mathcal{S}} \subseteq \mathcal{K} : |\widehat{\mathcal{S}}| = s\}$. Intuitively, imposing Condition 2 can reduce the average load, by dedicating more memory to a more popular file. Since the expression of the objective function in (4) is complex, unlike in [2], it is difficult to prove that Conditions 1 and 2 hold for the optimal file partition parameter under any arbitrary file popularity. Later, in Section V, the numerical results will verify that the two conditions are optimal properties. That is, imposing the two conditions will not lose optimality of Problem 1.

Next, we simplify Problem 1 under Conditions 1 and 2. First, we introduce some notations. Consider the number of representative users $|\underline{\mathcal{K}}(\mathbf{D})| = u$. Let $\widetilde{D}_{u,\langle 1 \rangle} \leq \widetilde{D}_{u,\langle 2 \rangle} \leq \ldots \leq \widetilde{D}_{u,\langle K-u \rangle}$ denote $(D_k)_{k \in \mathcal{K} \setminus \underline{\mathcal{K}}(\mathbf{D})}$ arranged in ascending order, so that $\widetilde{D}_{u,\langle i \rangle}$ is the i-th smallest. Let $P'_{i,u,n} \triangleq \Pr\left[\widetilde{D}_{u,\langle i \rangle} = n\right], \forall i = 1, \ldots, K-u$. By connecting the file request event $\widetilde{D}_{u,\langle i \rangle} = n$ to the "balls into bins problem", i.e., K balls are placed i.i.d. into N bins and each into bin n with probability p_n, we can obtain $P'_{i,u,n}$. The expression of $P'_{i,u,n}$ is given in [13], and is omitted here due to page limitation. Thus, by $P'_{i,u,n}$ and the simplification methods in [2] and [3], we have[2]

Lemma 1 (Simplification for Arbitrary File Popularity): Under Conditions 1 and 2, Problem 1 can be converted into:

Problem 2 (Simplified Problem of Problem 1):

$$\widetilde{R}^*_{\text{avg}}(K, N, M) \triangleq \min_{\mathbf{y}} \quad \widetilde{R}_{\text{avg}}(K, N, M, \mathbf{y})$$

$$s.t. \quad y_{n,s} \geq y_{n+1,s}, \quad \forall n \in \{1, 2, \ldots, N-1\}, s \in \{1, 2, \cdots, K\} \tag{7}$$

$$0 \leq y_{n,s} \leq 1, \quad \forall s \in \{0, 1, \cdots, K\},\ n \in \mathcal{N}, \tag{8}$$

$$\sum_{s=0}^{K} \binom{K}{s} y_{n,s} = 1, \quad \forall n \in \mathcal{N}, \tag{9}$$

$$\sum_{n=1}^{N} \sum_{s=1}^{K} \binom{K-1}{s-1} y_{n,s} \leq M, \tag{10}$$

where

$$\begin{aligned} &\widetilde{R}_{\text{avg}}(K, N, M, \mathbf{y}) \\ &\triangleq \sum_{s=1}^{K} \binom{K}{s} \sum_{n=1}^{N} \left(\left(\sum_{n'=n}^{N} p_{n'} \right)^s - \left(\sum_{n'=n+1}^{N} p_{n'} \right)^s \right) y_{n,s-1} \\ &- \sum_{u=1}^{\min\{K,N\}} \sum_{s=1}^{K-u} \binom{K-u}{s} \sum_{i=1}^{K-u} \binom{K-u-i}{s-1} \sum_{n=1}^{N} P'_{i,u,n} y_{n,s-1}. \end{aligned} \tag{11}$$

Problem 2 is a linear optimization problem with $N(K+1)$ variables and can be solved by using linear optimization techniques.

C. Optimization for Uniform Popularity without Subpacketization Constraint

In this part, we consider a special case, i.e., the uniform file popularity ($p_n = \frac{1}{N}$, for all $n \in \mathcal{N}$). First, we present another structural condition on the file partition parameter.

Condition 3 (Symmetry w.r.t. File): For all $n \in \{1, 2, \ldots, N-1\}$ and $s \in \{1, 2, \cdots, K\}$, when $p_n = p_{n+1}$,

$$y_{n,s} = y_{n+1,s}. \tag{12}$$

Condition 3 indicates that for all $n \in \{1, 2, \ldots, N-1\}$ and $s \in \{1, 2, \cdots, K\}$, when $p_n = p_{n+1}$, the size of subfiles $W_{n,\mathcal{S}}$, $\mathcal{S} \subseteq \{\widehat{\mathcal{S}} \subseteq \mathcal{K} : |\widehat{\mathcal{S}}| = s\}$ is the same as that of subfiles $W_{n+1,\mathcal{S}}$, $\mathcal{S} \subseteq \{\widehat{\mathcal{S}} \subseteq \mathcal{K} : |\widehat{\mathcal{S}}| = s\}$. Condition 3 ensures zero variance of the lengths of messages involved in the coded-multicast XOR operations, hence avoiding "bit waste" effect and increasing coded-multicasting opportunities. Later, we shall show that imposing this condition will not lose optimality of Problem 2 under the uniform file popularity.

By Condition 3, we can set

$$y_{n,s} = z_s, \quad \forall s \in \{0, 1, \cdots, K\},\ n \in \mathcal{N}. \tag{13}$$

[2]We omit all the proofs due to page limitation. Please refer to [13] for details.

Here, z_s can be viewed as the ratio between the number of data units in each subfile of type s of any file and the number of data units in any file (i.e., F). Let $\mathbf{z} \triangleq (z_s)_{s\in\{0,1,\cdots,K\}}$.

Next, we simplify Problem 1 under Conditions 1 and Condition 3. First, we introduce some notations. Let $P''_u \triangleq \Pr[|\underline{\mathcal{K}}(\mathbf{D})| = u]$, where $u \in \{1,2,\ldots,\min\{K,N\}\}$. Note that event $|\underline{\mathcal{K}}(\mathbf{D})| = u$ corresponds to the event in the "balls into bins" problem that there are u nonempty bins after placing K balls uniformly at random into N bins. By using results for the "balls into bins" problem in the uniform case [14] and considering Conditions 1 and 3, we have the following result.

Lemma 2 (Simplification for Uniform File Popularity): Under Conditions 1 and 3, Problem 1 can be converted into:

Problem 3 (Simplified Problem of Problem 1):

$$\min_{\mathbf{z}} \quad \sum_{s=0}^{K-1}\binom{K}{s+1}z_s - \sum_{u=1}^{\min\{K,N\}} P''_u \sum_{s=0}^{K-u-1}\binom{K-u}{s+1}z_s$$

$$s.t. \quad 0 \le z_s \le 1, \quad s \in \{0,1,\cdots,K\}, \tag{14}$$

$$\sum_{s=0}^{K}\binom{K}{s}z_s = 1, \tag{15}$$

$$\sum_{s=0}^{K}\binom{K}{s}s z_s \le \frac{KM}{N}. \tag{16}$$

where

$$P''_u = \left\{ {K \atop u} \right\} \frac{\binom{N}{u}u!}{N^K}, \tag{17}$$

and $\left\{ {K \atop u} \right\}$ is the Stirling number of the second kind.

Problem 3 is a linear optimization problem with $K+1$ variables and can be solved more efficiently than Problem 1.

Finally, we discuss the relation between an optimal solution of Problem 3 and Yu *et al.*'s centralized coded caching scheme [4].[3] Let $\mathbf{z}^* \triangleq (z_s^*)_{s\in\{0,1,\cdots,K\}}$ denote an optimal solution of Problem 3 and let $\widehat{R}^*_{\text{avg}}(K,N,M)$ denote the optimal value of Problem 3. Using KKT conditions, we have

Lemma 3 (Optimal Solution to Problem 3): For cache size $M \in \left\{0, \frac{N}{K}, \frac{2N}{K}, \ldots, N\right\}$, $\mathbf{z}^*$ is an optimal solution to Problem 3, where

$$z_s^* = \begin{cases} \frac{1}{\binom{K}{\frac{KM}{N}}}, & s = \frac{KM}{N} \\ 0, & s \in \{0,1,\cdots,K\} \setminus \{\frac{KM}{N}\}, \end{cases} \tag{18}$$

and the optimal value of Problem 3 is given by[4]

$$\begin{aligned} &\widehat{R}^*_{\text{avg}}(K,N,M) \\ =&\frac{K(1-M/N)}{1+KM/N} - \sum_{u=1}^{\min\{K,N\}} P''_u \binom{K-u}{\frac{KM}{N}+1}\Big/\binom{K}{\frac{KM}{N}}. \end{aligned} \tag{19}$$

[3]Yu *et al.*'s centralized coded caching scheme focuses on cache size $M \in \{0, \frac{N}{K}, \frac{2N}{K}, \ldots, N\}$, so that $\frac{KM}{N}$ is an integer in $\{0,1,\ldots,K\}$. For general $M \in [0,N]$, the worst-case load can be achieved by memory sharing.

[4]In this paper, we define $\binom{n}{k} = 0$ when $k > n$ [4].

Lemma 3 indicates that Yu *et al.*'s centralized coded caching scheme corresponds to an optimal solution of Problem 3. In addition, the optimal average load $\widehat{R}^*_{\text{avg}}(K,N,M)$ in (19) is equivalent to than that in [4]. Specifically, the first term in (19) corresponds to the worst-case load in [1] and the second term in (19) indicates the load reduction and is more tractable than that in [4]. Note that it has been shown that Yu *et al.*'s centralized coded caching scheme is optimal among all uncoded placement and all delivery under the uniform file popularity [4]. Thus, for the uniform file popularity, Conditions 1 and 3 are actually optimal properties.

IV. Average Load Minimization with Subpacketization Constraint

In this section, we minimize the average load by optimizing the file partition parameter under a subpacketization constraint which is given by

$$\|\mathbf{x}\|_0 \le \widehat{F}, \tag{20}$$

where $\|\mathbf{x}\|_0 \triangleq \sum_{n\in\mathcal{N}}\sum_{\mathcal{S}\in\mathcal{K}} \mathbf{1}[x_{n,\mathcal{S}} \neq 0] \in \{0,1,\ldots,N2^K\}$ denotes the ℓ_0-norm of the vector $\mathbf{x}$, i.e., the total number of subfiles for N files, and $\widehat{F} \in \{N, N+1, \ldots, N2^K\}$ represents the subpacketization limit. To the best of our knowledge, this is the first work explicitly considering a subpacketization constraint in optimization based coded caching design.

A. Problem Formulation

In this part, we minimize the average load under the file partition constraints in (1) and (2), the cache memory constraint in (3), and the subpacketization constraint in (20).

Problem 4 (Optimization with Subpacketization Constraint):

$$\begin{aligned} R^{\dagger}_{\text{avg}}(K,N,M) \triangleq \min_{\mathbf{x}} \quad & R_{\text{avg}}(K,N,M,\mathbf{x}) \\ s.t. \quad & (1),(2),(3),(20), \end{aligned}$$

where $R_{\text{avg}}(K,N,M,\mathbf{x})$ is given by (4).

Compared with Problem 1, Problem 4 has an extra constraint, i.e., the subpacketization constraint in (20). There are two main challenges in solving Problem 4. First, Problem 4 is an NP-Hard problem due to the discontinuous constraint in (20) involving the ℓ_0-norm [12]. Second, the number of variables in Problem 4 is $N2^K$, which is huge, especially when K and N are large.

B. Simplified Formulation

There are extensive research dealing with optimization problems involving the ℓ_0-norm. Those works can be divided into three main categories according to the way of treating the ℓ_0-norm, i.e., convex approximation, non-convex approximation, and non-convex exact reformulation [15]. For the category of convex approximation, one of the best known approaches is approximating the ℓ_0-norm with the ℓ_1-norm (denoted as $\|\mathbf{x}\|_1$) [15]. If the original optimization problem is convex except the discontinuous constraint involving the ℓ_0-norm, this convex approximation approach can transform the original NP-Hard problem involving the ℓ_0-norm into a convex problem.

However, it has been shown that an optimal solution of the approximated convex problem is not always sparse (may not be a feasible solution of the original problem) [16]. For the category of non-convex approximation, a variety of sparsity-inducing penalty functions, e.g., the ℓ_p pseudo-norm with $0 < p < 1$ [17], exponential concave function [18], and logarithmic function [19], have been proposed to approximate the ℓ_0-norm. In general, non-convex approximation can provide better sparsity than convex approximation, but may not provide an optimal solution of the original problem. Few works focus on non-convex exact reformulation, which is proposed to guarantee the equivalence between the reformulated problem and the original problem. Using exact penalty techniques, [15] and [20] show that the reformulated problems with suitable parameters are equivalent to the original problems. However, the reformulated problems are quite convoluted as they rely on too many parameters [12]. In recent work [12], the authors propose an exact DC (difference of two convex functions) reformulation which is simpler than the reformulated problems proposed in [15] and [20] and then use a DC algorithm to solve the DC problem. In the following, to obtain a simple equivalent problem of the original one, we use the exact DC reformulation method in [12].[5]

We first simplify Problem 4 to facilitate a low-complexity solution. Using the method for obtaining Problem 2 and Theorem 1 of [12] for simplifying the constraint in (20) under Conditions 1 and 2, we have the following result.

Lemma 4 (Simplification for for Arbitrary File Popularity): Under Conditions 1 and 2, Problem 4 can be converted into:

Problem 5 (Simplified Problem of Problem 4):

$$\begin{aligned} \widetilde{R}^{\dagger}_{\text{avg}}(K,N,M) \triangleq \min_{\mathbf{y}} \quad & \widetilde{R}_{\text{avg}}(K,N,M,\mathbf{y}) \\ s.t. \quad & (7),(8),(9),(10), \\ & \|W\mathbf{y}\|_{lgst,\widehat{F}} = N, \end{aligned} \tag{21}$$

where $\widetilde{R}_{\text{avg}}(K,N,M,\mathbf{y})$ is given by (11),

$$W \triangleq \begin{bmatrix} J & & & & \\ & J & & & \\ & & \cdot & & \\ & & & \cdot & \\ & & & & \cdot \\ & & & & & J \end{bmatrix} \tag{22}$$

denotes the block matrix of the dimension $N2^K \times (K+1)N$, with N blocks of matrix $J \triangleq (j_{m,n})_{m\in\{1,\ldots,2^K\},n\in\{1,\ldots,K+1\}}$ as its diagonal blocks and zero matrices as its non-diagonal blocks, and the element of row m and column n of J is

$$j_{m,n} = \begin{cases} 1, & \sum_{i=1}^{n-1}\binom{K}{i-1} < m \le \sum_{i=1}^{n}\binom{K}{i-1} \\ 0, & \text{otherwise} \end{cases}. \tag{23}$$

The number of variables in Problem 5 is $N(K+1)$, which is much smaller than that of Problem 4, i.e., $N2^K$. In addition, compared with Problem 2, Problem 5 has an extra constraint in (21), which is non-convex but has two advantages over the subpacketization constraint in (20): (i) $\|W\mathbf{y}\|_{lgst,\widehat{F}}$ in (21) is a convex function which will enable the transformation of the original problem into a DC programming problem that can be solved by a DC algorithm in polynomial time; (ii) a subgradient of $\|W\mathbf{y}\|_{lgst,\widehat{F}}$ can be efficiently computed, making the DC algorithm an efficient one. In Section IV-C, we will see the above two advantages clearly.

[5]Given the constraint in (9), the ℓ_1-norm of $\mathbf{x}$ is equal to a constant, i.e., $\|\mathbf{x}\|_1 = \sum_{i=1}^{N2^K} x_i = N$. Thus, replacing $\|\mathbf{x}\|_0$ with $\|\mathbf{x}\|_1$ in (20) results in the constraint $N \le \widehat{F}$, which always holds and cannot restrain $\mathbf{x}$.

C. Penalized Formulation and DC Algorithm

To facilitate the solution, we transform Problem 5 into a DC problem by using the exact DC reformulation method in [12].

Problem 6 (Penalized Formulation):

$$\begin{aligned} \min_{\mathbf{y}} \quad & \widetilde{R}_{\text{avg}}(K,N,M,\mathbf{y}) - \rho\|W\mathbf{y}\|_{lgst,\widehat{F}} \\ s.t. \quad & (7),(8),(9),(10), \end{aligned}$$

where $\widetilde{R}_{\text{avg}}(K,N,M,\mathbf{y})$ is given by (11) and the penalty parameter $\rho > 0$.

The objective function of Problem 6 is a difference of two convex functions. In addition, the constraints of Problem 6 are linear. Thus, Problem 6 is a DC problem and an efficient DC algorithm can be used to obtain a stationary point of Problem 6.

Let $\Upsilon \triangleq \{\mathbf{y} : (7),(8),(9),(10)\}$ and $\mathcal{V}(\Upsilon)$ be the vertex set of Υ. By Theorem 1 in [21], we show the equivalence between Problem 5 and Problem 6.

Lemma 5 (Exact Penalty): For all $\rho > \rho_0 \triangleq \frac{\widetilde{R}^{\dagger}_{\text{avg}}(K,N,M) - \widetilde{R}^{*}_{\text{avg}}(K,N,M)}{\min\left\{N - \|W\mathbf{y}\|_{lgst,\widehat{F}} : (7),(8),(9),(10), N > \|W\mathbf{y}\|_{lgst,\widehat{F}}\right\}}$, where $\widetilde{R}^{*}_{\text{avg}}(K,N,M)$ and $\widetilde{R}^{\dagger}_{\text{avg}}(K,N,M)$ are the optimal values of Problem 2 and Problem 5, respectively, Problem 5 and Problem 6 have the same optimal solution.

Lemma 5 shows that Problem 5 is equivalent to Problem 6 if the penalty parameter ρ is sufficiently large. Thus, in the following, we solve Problem 6 instead of Problem 5 by using a DC algorithm. The main idea of the DC algorithm is to iteratively solve a sequence of convex problems, each of which is obtained by linearizing the second term of the objective function of the DC problem. A subgradient of the second term of the objective function is required in the linearization in each iteration. Thus, to solve Problem 6, we first obtain a subgradient of $\|W\mathbf{y}\|_{lgst,\widehat{F}}$ by extending the closed-form expression of a subgradient of $\|\mathbf{y}\|_{lgst,\widehat{F}}$ given in [12].

Lemma 6 (Subgradient of $\|W\mathbf{y}\|_{lgst,\widehat{F}}$): $\mathbf{g}(\mathbf{y}) \triangleq (g_i)_{i\in\{1,2,\ldots,KN\}}$ is a subgradient of $\|W\mathbf{y}\|_{lgst,\widehat{F}}$, where

$$g_{[i]} = \begin{cases} \binom{K}{([i]-1) \mod (K+1)}, & i \in \{1,\ldots,I-1\} \\ \widehat{F} - \sum_{i=1}^{I-1}\binom{K}{([i]-1) \mod (K+1)}, & i = I \\ 0, & \text{otherwise} \end{cases}, \tag{24}$$

$[i]$ represents the index of i-th largest element in $\mathbf{y}$, and I satisfies $\sum_{i=1}^{I-1}\binom{K}{([i]-1) \mod (K+1)} \le \widehat{F}$ and $\sum_{i=1}^{I}\binom{K}{([i]-1) \mod (K+1)} > \widehat{F}$.

Then, based on the subgradient $\mathbf{g}(\mathbf{y})$ given in Lemma 6, we can obtain a stationary point of Problem 6 using the DC algorithm as summarized in Algorithm 2. Note that the obtained solution may not be a feasible solution of Problem 5. As in [15], to approach the globally optimal solution of Problem 5, we obtain multiple stationary points of Problem 6 by performing the DC algorithm multiple times, each with a random initial feasible point of Problem 6, and adopt the stationary point with the lowest average load among all the obtained stationary points of Problem 6 that are also feasible solutions of Problem 5.

Algorithm 2 DC Algorithm for Solving Problem 6

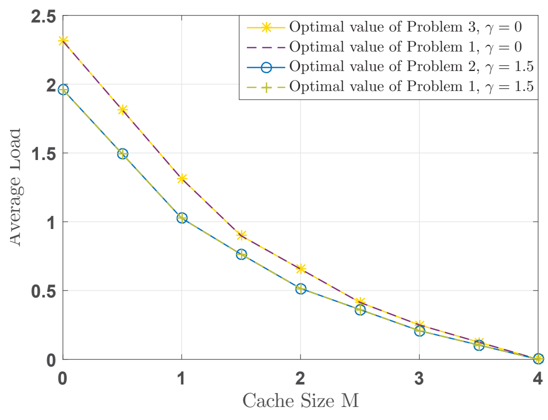
```
1: Find an initial feasible point $\mathbf{y}^{(0)}$ of Problem 6 and set $t = 0$
2: repeat
3:     Set $\mathbf{y}^{(t+1)}$ to be an optimal solution of the convex problem:
           $\min_{\mathbf{y}} \quad \widetilde{R}_{\text{avg}}(K,N,M,\mathbf{y}) - \rho \mathbf{g}\left(\mathbf{y}^{(t)}\right)^T \mathbf{y}$
           $s.t. \quad (7),(8),(9),(10)$
4:     Set $t = t + 1$
5: until $\left(\widetilde{R}_{\text{avg}}(K,N,M,\mathbf{y}^{(t-1)}) - \rho\|W\mathbf{y}^{(t-1)}\|_{lgst,\widehat{F}}\right) - \left(\widetilde{R}_{\text{avg}}(K,N,M,\mathbf{y}^{(t)}) - \rho\|W\mathbf{y}^{(t)}\|_{lgst,\widehat{F}}\right) \leq \delta$
```

V. Numerical Results

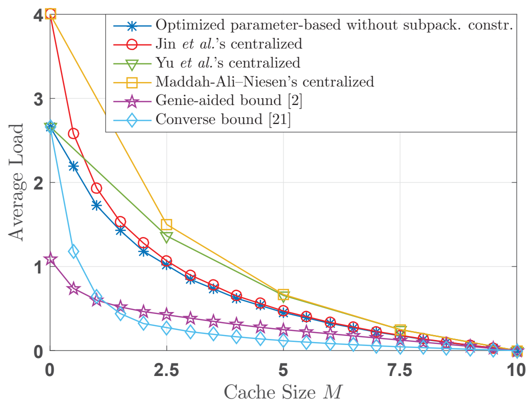
In the simulation, we assume the file popularity follows Zipf distribution, i.e., $p_n = \frac{n^{-\gamma}}{\sum_{n\in\mathcal{N}} n^{-\gamma}}$ for all $n \in \mathcal{N}$, where γ is the Zipf exponent. Fig. 2 (a) shows the optimal values of Problems 1, 2 and 3, verifying that Conditions 1, 2 and 3 are optimal conditions when the subpacktization constraint is not considered. Fig. 2 (b) compares the average load of our optimized parameter-based scheme, the average loads of the centralized coded caching schemes in [1], [2], [4], the genie-aided converse bound in [2] and the conserve bound in [22], all without considering the subpacketization issue. From Fig. 2 (b), we can see that the optimized parameter-based scheme outperforms the three baseline schemes. The gain over Jin *et al.*'s optimized centralized coded caching scheme follows by using an extended version of the improved delivery strategy of Yu *et al.*, that takes advantage of common requests (which occur with positive probability in the case of random demands). The gain over Yu *et al.*'s centralized coded caching scheme is due to exploiting the explicit knowledge of the file popularity in the optimization of content placement.[6] In addition, the optimized average load is close to the converse bounds, implying that the optimal value obtained by solving Problem 2 is close to optimal.

Fig. 3 (a) and Fig. 3 (b) compare the average load of our optimized parameter-based scheme considering the subpacketization constraint, the average loads of Tang *et al.*'s scheme [9] and Pareto-optimal PDA [11] both considering the subpacketization issue as well as the average load of our optimized parameter-based scheme without considering the subpacketization constraint (serving as a lower bound of the proposed one considering the subpacketization constraint). Note that in Fig. 3 (a), for the considered K, N, M, Tang *et al.*'s scheme is applicable only at $\widehat{F} = 20$ and Pareto-optimal PDA is applicable only at $\widehat{F} = 20$ and $\widehat{F} = 100$; in Fig. 3 (b), for the considered K, N, M, Tang *et al.*'s scheme and Pareto-optimal PDA are not applicable at any $\widehat{F}$. From Fig. 3 (a) and Fig. 3 (b), we can see that our optimized parameter-based scheme outperforms Tang *et al.*'s scheme and Pareto-optimal PDA in terms of both the average load and application region. In addition, from Fig. 3 (b), we can see that our optimized parameter-based scheme considering the subpacketization constraint can achieve significantly lower subpacketization than the one without considering the subpacketization constraint, at the cost of a small increase of the average load. This means

[6]It has been proved in [2] that the optimized parameter-based scheme in [2] outperforms Maddah-Ali–Niesen's centralized coded caching scheme [1].

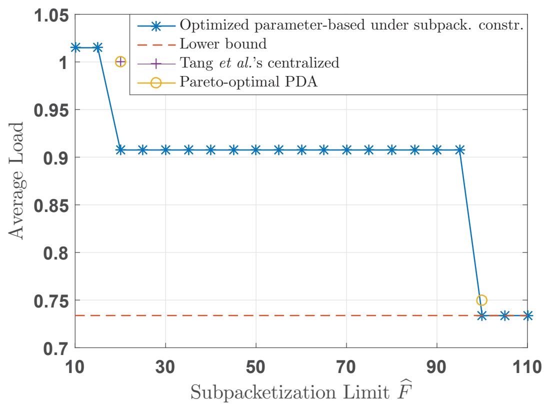


(a) $K = 3$, $N = 4$.

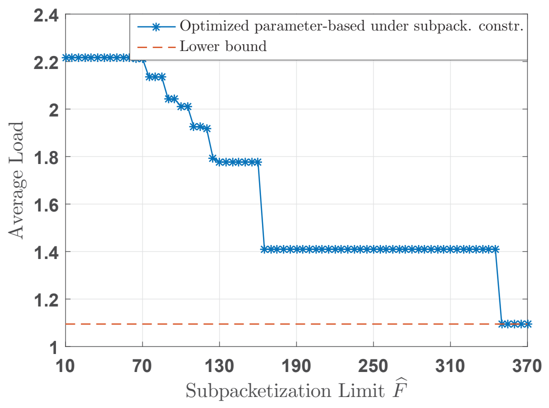


(b) $K = 4$, $N = 10$, $\gamma = 1.5$.

Fig. 2: Average load versus M under Zipf distribution. Note that Maddah-Ali–Niesen's and Yu *et al.*'s centralized coded caching schemes mainly focus on the cache size $M \in \left\{0, \frac{N}{K}, \frac{2N}{K}, \ldots, N\right\}$. For other $M \in [0, N]$, the average loads of Maddah-Ali–Niesen's and Yu *et al.*'s centralized coded caching schemes are achieved by memory sharing [1], [4].



(a) $K = 6$, $N = 5$, $M = 2.5$, $\gamma = 1.3$.



(b) $K = 6$, $N = 10$, $M = 4$, $\gamma = 0.5$.

Fig. 3: Average load versus $\widehat{F}$ under Zipf distribution. To obtain the average load of our optimized parameter-based scheme under the subpacketization constraint, we solve Problem 6 with $\rho = 500$ by performing Algorithm 2 with $\delta = 0.001$ 100 times each with a random initial feasible point and adopt the stationary point with the lowest average load among all the obtained stationary points of Problem 6 that are also feasible solutions of Problem 5.

that sacrificing a small average load gain can achieve a huge subpacketization reduction under the considered setting.

VI. Conclusion

In this paper, we first presented a class of centralized coded caching schemes consisting of a general content placement strategy specified by a file partition parameter, enabling efficient and flexible content placement, and a specific content delivery strategy, enabling load reduction by exploiting common requests of different users. Then, we considered two cases: the case without considering the subpacketization issue and the case considering the subpacketization issue. In the first case, we formulated the coded caching optimization problem over the considered class of schemes with $N2^K$ variables to minimize the average load under an arbitrary file popularity. Imposing some conditions on the file partition parameter, we transformed the original optimization problem into a linear optimization problem with $N(K+1)$ variables under an arbitrary file popularity and a linear optimization problem with $K+1$ variables under the uniform file popularity. In the second case, we first formulated the coded caching optimization problem over the considered class of schemes to minimize the average load under an arbitrary file popularity subject to a subpacketization constraint involving the ℓ_0-norm. Next, imposing the same conditions and using the exact DC reformulation method, we converted the original problem with $N2^K$ variables into a simplified DC problem with $N(K+1)$ variables. Then, we used a DC algorithm to solve the simplified DC problem.

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] S. Jin, Y. Cui, H. Liu, and G. Caire, “Structural properties of uncoded placement optimization for coded delivery,” *CoRR*, vol. abs/1707.07146, 2017.
[3] A. M. Daniel and W. Yu, “Optimization of heterogeneous coded caching,” *CoRR*, vol. abs/1708.04322, 2017.
[4] Q. Yu, M. A. Maddah-Ali, and A. S. Avestimehr, “The exact rate-memory tradeoff for caching with uncoded prefetching,” *CoRR*, vol. abs/1609.07817, 2016.
[5] K. Shanmugam, M. Ji, A. M. Tulino, J. Llorca, and A. G. Dimakis., “Finite-length analysis of caching-aided coded multicasting,” *IEEE Trans. Inf. Theory*, vol. 62, no. 10, pp. 5524–5537, Oct. 2016.
[6] S. Jin, Y. Cui, H. Liu, and G. Caire, “New order-optimal decentralized coded caching schemes with good performance in the finite file size regime,” *CoRR*, vol. abs/1604.07648, 2016.
[7] M. Ji, K. Shanmugam, G. Vettigli, J. Llorca, A. M. Tulino, and G. Caire, “An efficient multiple-groupcast coded multicasting scheme for finite fractional caching,” in *IEEE ICC*, Jun. 2015, pp. 3801–3806.
[8] K. Wan, D. Tuninetti, and P. Piantanida, “Novel delivery schemes for decentralized coded caching in the finite file size regime,” in *IEEE ICC Workshops*, May 2017, pp. 1183–1188.
[9] L. Tang and A. Ramamoorthy, “Coded caching with low subpacketization levels,” in *IEEE Globecom Workshops*, Dec. 2016, pp. 1–6.
[10] K. Shanmugam, A. M. Tulino, and A. G. Dimakis, “Coded caching with linear subpacketization is possible using ruzsa-szeméredi graphs,” in *IEEE ISIT*, Jun. 2017, pp. 1237–1241.
[11] M. Cheng, Q. Yan, X. Tang, and J. Jiang, “Coded caching schemes with low rate and subpacketizations,” *CoRR*, vol. abs/1708.06650, 2017.
[12] J.-y. Gotoh, A. Takeda, and K. Tono, “Dc formulations and algorithms for sparse optimization problems,” *Math Programming*, pp. 1–36, 2017.
[13] S. Jin, Y. Cui, H. Liu, and G. Caire, “Uncoded placement optimization for coded delivery,” *CoRR*, vol. abs/1709.06462, 2017. [Online]. Available: http://arxiv.org/abs/1709.06462
[14] P. Flajolet and R. Sedgewick, *Analytic combinatorics*, 2009.
[15] H. A. Le Thi, T. P. Dinh, H. M. Le, and X. T. Vo, “Dc approximation approaches for sparse optimization,” *European Journal of Operational Research*, vol. 244, no. 1, pp. 26–46, 2015.
[16] S. Shalev-Shwartz, N. Srebro, and T. Zhang, “Trading accuracy for sparsity in optimization problems with sparsity constraints,” *SIAM Journal on Optimization*, vol. 20, no. 6, pp. 2807–2832, 2010.
[17] W. Fu, “Penalized regressions: The bridge versus the lasso,” *Journal of Computational and Graphical Statistics*, pp. 397–416, 1998.
[18] P. S. Bradley and O. L. Mangasarian, “Feature selection via concave minimization and support vector machines.” in *ICML*, vol. 98, 1998.
[19] J. Weston, A. Elisseeff, B. Schölkopf, and M. Tipping, “Use of the zero norm with linear models and kernel methods,” *J. Mach. Learn. Res.*, vol. 3, pp. 1439–1461, Mar. 2003.
[20] T. Pham Dinh and H. A. Le Thi, “Recent advances in dc programming and dca,” in *Trans on Computational Intelligence*, 2014, pp. 1–37.
[21] H. A. Le Thi, T. P. Dinh, and H. Van Ngai, “Exact penalty and error bounds in dc programming,” *Journal of Global Optimization*, vol. 52, no. 3, pp. 509–535, 2012.
[22] C. Wang, S. H. Lim, and M. Gastpar, “A new converse bound for coded caching,” *CoRR*, vol. abs/1601.05690, 2016.