This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIT.2019.2946361, IEEE Transactions on Information Theory

Full Characterization of Optimal Uncoded Placement for the Structured Clique Cover Delivery of Nonuniform Demands

Seyed Ali Saberali, Lutz Lampe and Ian Blake

***Abstract*—We investigate the problem of coded caching for nonuniform demands when the structured clique cover algorithm proposed by Maddah-Ali and Niesen for decentralized caching is used for delivery. We apply this algorithm to all user demands regardless of their request probabilities. This allows for coding among the files that have different request probabilities but makes the allocation of memory to different files challenging during the content placement phase. As our main contribution, we analytically characterize the optimal placement strategy that minimizes the expected delivery rate under a storage capacity constraint. It is shown that the optimal placement follows either a two or a three group strategy, where a set of less popular files are not cached at all and the files within each of the other sets are allocated identical amounts of storage as if they had the same request probabilities. We show that for a finite set of storage capacities, that we call the base-cases of the problem, the two group strategy is always optimal. For other storage capacities, optimal placement is achieved by memory sharing between certain base-cases and the resulting placement either follows a two or a three group strategy depending on the corresponding base-cases used. We derive a polynomial time algorithm that determines the base-cases of the problem given the number of caches and popularity distribution of files. Given the base-cases of the problem, the optimal memory allocation parameters for any storage capacity are derived analytically.**

I. Introduction

A. Background

The next generation wireless communication networks deploy a dense composition of short-range and low-power small-cells and combine them with the macrocells into heterogeneous networks. This architecture promotes localized communications and effectively increases the area spectral efficiency of the network. The performance of such networks is however challenged by the congestion of the backhaul links that connect the small-cells to the backbone communications network during the peak traffic hours. Caching at the edge is a promising technique to alleviate the backhaul congestion through the storage of popular content closer to the end users [1]–[8].

Coded caching [8]–[10] is a novel approach for content caching in a network that consists of multiple caching nodes which communicate with a central server over a shared broadcast channel. This technique benefits from network coding and coded multicasting to gain superlinear reduction in the data delivery load on the shared link as the cache capacity increases. In particular, during a placement phase, popular content is carefully distributed over the different storage nodes such as to create coding opportunities among the caches. During a delivery phase, the content that is requested but is missing from the caching nodes is delivered to them by the central server's transmissions over the shared link. The server exploits the coding opportunities created during placement to embed the missing content requested by multiple caches in a single message that every target cache can decode for its desired content. The load on the shared link is referred to as the delivery rate.

The coded caching proposed by Maddah-Ali and Niesen in their seminal work [10] efficiently utilizes the side information that each cache has about the requests of the other caches in order to build server's coded messages. This delivery algorithm can be viewed as a structured way of clique-covering the vertices of the side information graph that characterizes the availability of the content requested by each cache in the other caches. As a result, we call this delivery algorithm Structured Clique Cover (SCC) procedure throughout this paper.

The seminal works [9] and [10] aimed at minimizing the peak delivery rate when different files in the library are equally likely to be requested by users, i.e., when the user demands are uni-

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIT.2019.2946361, IEEE Transactions on Information Theory

form. However, a more practical scenario concerns caching of files with different popularity levels. In this scenario, it is expected to allocate more storage to the caching of the more popular files during placement. This idea is followed in several works in the literature [11]–[16].

B. Related Work

Two major approaches are followed for coded caching with nonuniform demands. The first approach is based on the grouping of files into different popularity groups based on their request probabilities [11], [12], [14]. With the files in each group having relatively similar request probabilities, the SCC algorithm is applied separately to the requests belonging to each group for delivery. The advantage of this method is the simplicity of the analysis of the expected rate. Its main disadvantage is that it limits the utilization of coding opportunities to the files that are within each popularity group. The design objective in this approach is to find the grouping of files that achieves the lowest expected rate. A higher number of groups provides higher degrees of freedom to assign different amounts of storage to files with different request probabilities. On the other hand, the larger the number of groups is, the more underutilized are the coding opportunities among the different groups. In [14], the library is grouped into two categories of popular and unpopular files. The requests for popular files are delivered by the SCC algorithm while the requests of unpopular files are delivered through uncoded messages. This is an extreme case of the grouping approach and its expected rate is shown to be at most a constant factor away from the information theoretic lower bound, independent of the file popularity distribution.

The second approach is followed in [15] and [16] which applies the SCC algorithm to all the user demands regardless of their request probabilities and the amount of storage allocated to each file. For any fixed placement of content, this delivery scheme outperforms the previously discussed group-based delivery. However, optimization of the amount of memory allocated to each file is challenging because of the complicated interplay between the memory allocation parameters and the expected delivery rate. References [15] and [16] use a convex optimization formulation of the memory allocation problem which aims to minimize the expected delivery rate for a given storage capacity per cache. We refer to this problem as Rate Minimization with Storage Constraint (RMSC). Reference [16] has followed a numerical approach to solve the RMSC problem and is mainly focused on reducing the computational complexity of the numerical analysis involved. In contrast, [15] follows a theoretical approach to find structural properties in the optimal solution of the problem.

C. Our Contributions

The results provided in [15] do not capture specific properties of the optimal solution which can considerably facilitate solving the memory allocation problem. In this work, we find such structures in the optimal solution and solve the RMSC problem analytically when user demands are nonuniform and the SCC procedure is used for delivery. In particular, we will show that such properties enable the derivation of the optimal solution based on a search over a finite set of points. The cardinality of this set scales linearly with the product of the number of caches and the number of files. The properties that we derive also provide a unifying interpretation of the optimal placement strategy for both uniform and nonuniform popularity distribution of files, as we will discuss in the remainder of this section.

As the first structural property, we show that for instances of the problem with cache capacities that belong to a finite set $\mathcal{M}$, the optimal placement for RMSC follows the simple pattern of splitting the library of files into two groups. One group consists of less popular files and the files in this group are not cached at all. The files in the second group are cached but are treated as if they had the same request probabilities. We call such instances of RMSC the *base-cases*.

For instances of the problem that are not among the base-cases, we prove that the optimal solution is achieved by a convex combination of the solutions to certain base-cases of the problem. This solution is identical to the placement parameters obtained by memory sharing between the two base-cases of the RMSC problem. Memory sharing is already shown to be optimal when demands are uniform [15, Lemma 5], [16, Theorem 1], [17, Proposition 1]. Hence, this result shows that memory sharing is also optimal for nonuniform demands when applied to the appropriately chosen instances of the problem. To elaborate, let K, N and M be the number of caches, files in the library and files that each cache

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIT.2019.2946361, IEEE Transactions on Information Theory

can store, respectively. For optimal placement of identically popular files when SCC delivery is used, we have the following [15]–[17]:

- All files are treated identically during placement, in particular, the same amount of storage is allocated to the caching of each file.
- For a cache size M that corresponds to an integer value of $t = K\frac{M}{N}$, the optimal placement breaks each file into $\binom{K}{t}$ nonoverlapping segments. Then, it exclusively stores each one of the segments in exactly one of the $\binom{K}{t}$ subsets of caches that have cardinality t. We refer to these cases of the problem as the base cases and denote by $\mathcal{M}$ the set of corresponding cache sizes $\{0, \frac{1}{K}N, \frac{2}{K}N, \dots, N\}$.
- For other cache capacities, the optimal placement can be obtained by memory sharing between the optimal placements for two instances of the problem with cache capacities $M_l = \max\{m \in \mathcal{M} \mid m < M\}$ and $M_u = \min\{m \in \mathcal{M} \mid M < m\}$.[1]

We prove that a similar pattern exists in the optimal placement for nonuniform demands. In particular, we propose an algorithm with worst-case complexity of $O(K^2N^2)$ to derive the set $\mathcal{M}$ given a nonuniform popularity distribution for the files. If $M \notin \mathcal{M}$, the optimal placement is obtained by memory sharing between $M_l, M_u \in \mathcal{M}$ as it was done for uniform demands using the derived set $\mathcal{M}$. In this case, optimal placement either follows a two or a three group strategy depending on the specifics of the two corresponding base-cases used.

For the optimal placement that we derive, the memory allocated to different files does not show a gradual and smooth increase as the request probability increases. Instead, for base-cases where the two-group strategy is optimal, the memory allocation exhibits a binary behavior, i.e., as the request probability increases the amount of memory allocated to the files shows an abrupt increase from zero to a new level at a certain request probability and remains at that level thereafter. A similar trend exists for non base-cases, but there might be two thresholds on request probabilities where the jumps in the memory allocated to files occur.

[1]The idea of memory sharing for uniform demands was presented in [9] as an achievable scheme when t is not an integer. References [15]–[17] proved that memory sharing is optimal for SCC delivery when demands are uniform.

Finally, we find the results in [14] closely connected to the results that we derive in this paper. Reference [14] considers the setting of randomized placement algorithms and within that setting, it shows that a two-group (or occasionally a three-group) strategy guarantees a delivery rate within a constant factor of the information theoretic lower bound. In this work, we derive a deterministic placement of files in the caches which solves the RMSC problem, i.e., we analytically prove its optimality when the SCC delivery algorithm is applied to all user demands. We prove that the expected delivery rate of our optimal solution is a lower-bound on the expected delivery rates of the group-based methods that apply the SCC algorithm within each group of requested files for delivery. Given that the coded caching in [14] follows such a scheme and its rate is within a constant factor from the information-theoretic lower bound, it concludes that the delivery rate of RMSC is also within a constant factor of the optimum delivery rate. Through numerical examples we show that the grouping of files given by these two schemes and the delivery rates resulting from them can be significantly different for specific regimes of problem parameters. Further, we compare the expected rate of RMSC to the information-theoretic outer-bound on the expected rate of caching schemes with uncoded prefetching derived in [18]. This comparison suggests that the expected rate of RMSC approaches the outer-bound as the cache size increases. We provide a detailed discussion to show that the existence of this performance gap is, at least partially, due to an inefficiency in the SCC algorithm for delivery. We suggest directions for future research in order to reduce or fully close this performance gap.

The remainder of this paper is organized as follows. The setup of the problem and formulation of the expected rate and the storage used in terms of placement parameters are presented in Section II. For better readability of the paper, a list of the symbols and acronyms that we use in the paper is provided in Table I. The RMSC problem is formulated in Section III and a duality framework is proposed for it. Structures in the optimal solution of RMSC for the base-cases are derived in Section IV. In Section V, we propose an algorithm to identify the base-cases of the RMSC problem for any given popularity distribution of files and we derive the optimal solution of RMSC. We conclude the paper

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIT.2019.2946361, IEEE Transactions on Information Theory

TABLE I. List of Symbols and Acronyms

RMSC	Rate Minimization with Storage Constraint
JRSM	Joint Rate and Storage Minimization
SCC	Structured Clique-Cover algorithm
K	number of caches
N	number of files
M	cache capacity (files)
F	length of files (bits)
R	delivery rate (files)
p_n	request probability of file n
$\mathcal{P}$	a placement of files in caches
A_D	a delivery algorithm
$[n]$	set $\{1,\dots,n\}$
$[n]_i$	set $\{i+1,i+2,\dots,i+n\}$
$\mathcal{S}$	a subset of $[K]$
X^n	set of the bits of file n
$X^n_{\mathcal{S}}$	set of bits of file n that are exclusively cached in caches in $\mathcal{S}$
$x^n_{\mathcal{S}}$	length of subfile $X^n_{\mathcal{S}}$ normalized by F bits
x^n_s	$x^n_{\mathcal{S}}$ for $\mathcal{S}:\lvert\mathcal{S}\rvert=s$
y^n_s	$\binom{K}{s}x^n_s$
d_k	index of file demanded by cache k
$\boldsymbol{d}$	demand vector $(d_1,\dots,d_K)$
$\boldsymbol{d}_{\mathcal{S}}$	subdemand vector for requests of caches in $\mathcal{S}$
$\mathcal{G}_s$	set of all subsets of $[N]$ with cardinality less than or equal s
π^g_s	probability that for a set of caches $\mathcal{S}:\lvert\mathcal{S}\rvert=s$, $g\in\mathcal{G}_s$ is the set of files in $\boldsymbol{d}_{\mathcal{S}}$
$\mathcal{M}$	set of cache capacities for base-cases of RMSC
$\mathcal{R}$	set of optimal rates for base-case of RMSC
$\mathcal{Y}^*$	set of optimal solutions for base-cases of RMSC
M_l	$\max\{m\in\mathcal{M}\mid m<M\}$
M_u	$\min\{m\in\mathcal{M}\mid M<m\}$
Supp($\boldsymbol{w}$)	support of vector $\boldsymbol{w}$
γ	Lagrange multiplier for capacity constraint

in Section VI.

II. Problem Setup and Formulation of Utilized Storage and Expected Delivery Rate

A. Problem Setup

We consider the canonical network of multiple caches and a central server as modeled in [11] for the general case where user demands can be nonuniform (see Fig. 1). In particular, we consider a library of N files, each of length F bits and a network of K caches, each with storage capacity of MF bits. All files are available in the central server and the server can communicate with the caching nodes over an error-free broadcast link.

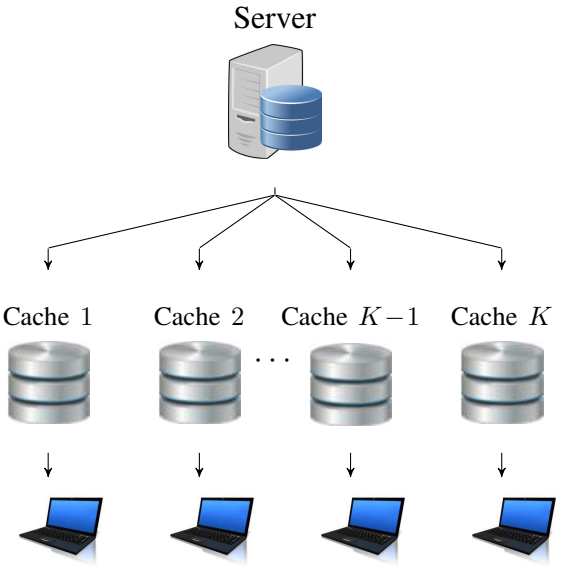


Fig. 1. A network with K caches and a central server.

Notation 1: We use notation $[n]$ to denote the set of the first n positive integers $\{1,\dots,n\}$. Similarly, we use $[n]_i$ to denote the set of the first n positive integers larger than i, i.e., $\{i+1,i+2,\dots,i+n\}$.

The placement of files in the caches can be described as follows. Let X^n be the set of the bits of file n. Then, for each $\mathcal{S}\subset[K]$, set $X^n_{\mathcal{S}}\subset X^n$ represents the bits of file n that are exclusively cached in caches in $\mathcal{S}$.[2] By definition, subsets $X^n_{\mathcal{S}}$ are disjoint for different $\mathcal{S}$ and $\bigcup_{\mathcal{S}\subset[K]}X^n_{\mathcal{S}}=X^n$. Also, define $x^n_{\mathcal{S}}=|X^n_{\mathcal{S}}|/F$ as the ratio of bits of file n that are exclusively cached in the caches in $\mathcal{S}$. Then, it follows that $\sum_{\mathcal{S}\subset[K]}x^n_{\mathcal{S}}=1$ for every $n\in[N]$. We denote by $\boldsymbol{x}$ the vector of all placement parameters $x^n_{\mathcal{S}}$.

The server is informed of all caches' content. For a fixed placement of files in caches, every cache $k\in[K]$ reveals one request for a file $d_k\in[N]$ at each time instant. We refer to $\boldsymbol{d}=[d_1,\dots,d_K]$ as the demand vector which represents the demands of all caches at the current time instant. Similarly, for a subset of caches $\mathcal{S}$, subdemand vector $\boldsymbol{d}_{\mathcal{S}}$ determines the files requested by the caches in $\mathcal{S}$ in the same order as in $\boldsymbol{d}$.

We assume that requests for different files are independent and the request probabilities do not change for different caches. Let $\{p_n\}_{n\in[N]}$ represent the request probabilities of the files. Here, files are

[2]In other words, bits $X^n_{\mathcal{S}}\subset X^n$ are stored in every cache in $\mathcal{S}$ and are not stored in any cache in $[K]\setminus\mathcal{S}$.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIT.2019.2946361, IEEE Transactions on Information Theory

Algorithm 1 Delivery by SCC [10]

```
procedure DELIVERY(d; {X^n}_{n=1,...,N})
    for s = 1, ..., K do
        for S ⊂ [K] : |S| = s do
            server sends ⊕_{k∈S} X^{d_k}_{S\k}
        end for
    end for
end procedure
```

sorted in the decreasing order of request probabilities, i.e., $n > m$ implies $p_n \leq p_m$. We refer to the file request probabilities as popularity distribution. For a demand vector $\boldsymbol{d}$ and every $k \in [K]$, the parts of file d_k that are available in cache k are locally delivered to its user. The missing parts are provided by the server over the broadcast channel through a signal of size $R(\boldsymbol{d}; \mathcal{P}, A_D)$ files. The quantity $R(\boldsymbol{d}; \mathcal{P}, A_D)$ is the delivery rate measured in the equivalent number of files for the demand vector $\boldsymbol{d}$, given a specific placement of files $\mathcal{P}$ and a delivery algorithm A_D. Placement $\mathcal{P}$ is fixed for all the demand vectors that arrive during the delivery phase. It is required that every cache that has forwarded its request to the server be able to decode the broadcasted signal for the content it requested. We are interested in minimizing $\mathbb{E}_{\boldsymbol{d}}\left(R(\boldsymbol{d}; \mathcal{P}, A_D)\right)$, where the expectation is over the randomness in the demand vector $\boldsymbol{d}$.

B. Delivery Algorithm

In this work, we apply the SCC procedure to all user demands for delivery regardless of their popularity levels and the memory allocated to them. The delivery procedure is shown in Algorithm 1. By following Algorithm 1, the server transmits messages of the form

$$\oplus_{k\in\mathcal{S}} X^{d_k}_{\mathcal{S}\backslash k} \tag{1}$$

for every nonempty $\mathcal{S} \subset [K]$. All the components $X_{\mathcal{S}\backslash k}$ embedded in the message are zero-padded to the length of the largest component. Hence, the length of the message is $\max_{k\in\mathcal{S}} |X^{d_k}_{\mathcal{S}\backslash k}|$.[3]

[3]From a graph theoretic perspective, this message corresponds to XORing the requested subfiles that form a clique in the side information graph [19, Section II.A] and [20, Section I.A]. Since the set of messages $\oplus_{k\in\mathcal{S}} X^{d_k}_{\mathcal{S}\backslash k}$ delivers all the missing subfiles, it covers all the vertices in the side information graph. Hence, one can see the delivery procedure of [10] as a structured way of covering the side information graph vertices with cliques.

As mentioned in Section I-B, Algorithm 1 contrasts the delivery schemes in [11], [14], [21] which are also based on the SCC procedure but separately apply it to the files with *close* request probabilities. Algorithm 1 has the advantage that it allows coding among all files regardless of their request probabilities and can result in a smaller delivery rate. To elaborate, message (1) delivers every subset of bits in $\{X^{d_k}_{\mathcal{S}\backslash k}\}_{k\in\mathcal{S}}$ to the corresponding requesting cache in $\mathcal{S}$. Given a grouping of files into groups $l = 1, \ldots, L$, if instead of applying the SCC to the whole demand vector we applied it to subdemand vectors consisting of files in the same popularity group, the exact same subfiles delivered by (1) would have been delivered through the set of messages $\left\{\oplus_{k\in\hat{\mathcal{S}}_l} X^{d_k}_{\mathcal{S}\backslash k}\right\}_{l=1}^{L}$ where $\hat{\mathcal{S}}_l = \{k \in \mathcal{S} | d_k \in l\text{-th group}\}$. This message has length $\sum_{l=1}^{L} \max_{k\in\hat{\mathcal{S}}_l} |X^{d_k}_{\mathcal{S}\backslash k}|$ which is lower bounded by $\max_{k\in\mathcal{S}} |X^{d_k}_{\mathcal{S}\backslash k}|$ which is the length of (1) with SCC applied to the whole demand vector. A direct consequence of this argument is that with an optimal placement for Algorithm 1, its delivery rate would be a lower-bound on the delivery rates of caching schemes like the ones in [11], [14], [21] which apply Algorithm 1 to subdemand vectors that consist of files that are in identical popularity groups. In particular, the fact that the delivery rate of [14] is within a constant factor of the information-theoretic lower-bound [14, Section III] implies that the delivery rate of Algorithm 1 with the optimal placement that we derive here is also within a constant factor of the information-theoretic minimum rate.

C. Formulation of Expected Delivery Rate and Storage

To derive optimal placement for SCC delivery, we need to formulate the expected delivery rate and the storage used by the placement algorithm in terms of the placement parameters $x^n_{\mathcal{S}}$.

Expected Delivery Rate: For Algorithm 1 as the delivery algorithm, the delivery load is

$$R(\boldsymbol{d}; \boldsymbol{x}) = \sum_{\substack{\mathcal{S}:\mathcal{S}\subset[K]\\ \mathcal{S}\neq\emptyset}} \max_{k\in\mathcal{S}} x^{d_k}_{\mathcal{S}\backslash k}$$

for a given demand vector $\boldsymbol{d}$ and placement parameters $x^n_{\mathcal{S}}$. To formulate the expected delivery rate in terms of the placement parameters, let $R_{\mathcal{S}}(\boldsymbol{d}; \boldsymbol{x}) =$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIT.2019.2946361, IEEE Transactions on Information Theory

$\max_{k\in\mathcal{S}} x^{d_k}_{\mathcal{S}\backslash k}$ denote the rate required to deliver the subfiles that are exclusively stored in the subset of caches $\mathcal{S}$. Then, the expected rate with respect to randomness in user demands is

$$r(\boldsymbol{x}) \triangleq \mathbb{E}_{\boldsymbol{d}}\left(R(\boldsymbol{d};\boldsymbol{x})\right) = \sum_{\substack{\mathcal{S}:\mathcal{S}\subset\mathcal{K}\\ \mathcal{S}\neq\emptyset}} \mathbb{E}_{\boldsymbol{d}}(R_{\mathcal{S}}(\boldsymbol{d};\boldsymbol{x})).$$

We assumed that the popularity distribution of files is the same for different caches. We use this symmetry in the demand probabilities of the different caches to simplify the placement formulation by setting $x^n_{\mathcal{S}} = x^n_s$ for all $\mathcal{S} : |\mathcal{S}| = s$. In other words, for a given file, the portion of bits that is exclusively cached in any subset of caches $\mathcal{S}$ with cardinality s is the same.

Proposition 1: The assumption $x^n_{\mathcal{S}} = x^n_s$ for all $\mathcal{S} : |\mathcal{S}| = s$ is without loss of optimality for the RMSC problem.

Proof: See Appendix A. ∎

Because of the symmetric structure of the placement, $\mathbb{E}_{\boldsymbol{d}}(R_{\mathcal{S}}(\boldsymbol{d};\boldsymbol{x}))$ is the same for all $\mathcal{S} : |\mathcal{S}| = s$, and it can be denoted by $\bar{R}_s(\boldsymbol{x})$. Hence, the average rate can be written as

$$r(\boldsymbol{x}) = \sum_{s=1}^{K} \binom{K}{s} \bar{R}_s(\boldsymbol{x}).$$

Let $\mathcal{G}_s$ be the set of all subsets of $[N]$ with cardinality at most s. Let π^g_s denote the probability that files $g \in \mathcal{G}_s$ be requested by a set of caches $\mathcal{S}$ with $|\mathcal{S}| = s$. Then,

$$\bar{R}_s(\boldsymbol{x}) = \sum_{g\in\mathcal{G}_s} \pi^g_s \max_{n\in g} x^n_{s-1}$$

and therefore, the expected delivery rate is

$$\sum_{s=1}^{K} \binom{K}{s} \sum_{g\in\mathcal{G}_s} \pi^g_s \max_{n\in g} x^n_{s-1},$$

which can equivalently be written as

$$\sum_{s=0}^{K-1} \binom{K}{s+1} \sum_{g\in\mathcal{G}_{s+1}} \pi^g_{s+1} \max_{n\in g} x^n_s.$$

Storage Used by Placement: The total storage used by cache $k \in [K]$ is

$$\sum_{n=1}^{N} \sum_{\substack{\mathcal{S}\subset[K]:\\ k\in\mathcal{S}}} x^n_{\mathcal{S}}, \tag{2}$$

where under the symmetry condition $x^n_{\mathcal{S}} = x^n_s$ for all $\mathcal{S} : |\mathcal{S}| = s$, this quantity simplifies to

$$\sum_{n=1}^{N} \sum_{s=1}^{K} \binom{K-1}{s-1} x^n_s$$

for every cache. The inner sum is the storage that is assigned to file n in each cache, as for each file n, each cache k stores the subfiles $X^n_{\mathcal{S}} : k \in \mathcal{S}$. There are $\binom{K-1}{s-1}$ subsets of $[K]$ of cardinality s with this property for each file. The outer sum adds up the storage used for all the files in the library.

Change of Variable for Placement Parameters: For simpler exposition of the optimization problems and better interpretability of the results, we find it useful to use the change of variable

$$y^n_s = \binom{K}{s} x^n_s. \tag{3}$$

Variable y^n_s is the total portion of bits of file n that is cached exclusively in all the $\binom{K}{s}$ different subsets of $[K]$ with cardinality s. As argued in Section II-A, we have $\sum_{\mathcal{S}\subset[K]} x^n_{\mathcal{S}} = 1$. Given that $x^n_{\mathcal{S}} = x^n_{|\mathcal{S}|}$ and using the change of variable (3), it follows that

$$\sum_{s=0}^{K} y^n_s = 1, \qquad \forall n \in [N]. \tag{4}$$

As a result, the expected rate and storage can be formulated as functions of the new placement parameters y^n_s as

$$r(\boldsymbol{y}) = \sum_{s=0}^{K-1} \frac{K-s}{s+1} \sum_{g\in\mathcal{G}_{s+1}} \pi^g_{s+1} \max_{n\in g} y^n_s, \tag{5}$$

$$m(\boldsymbol{y}) = \sum_{n=1}^{N} \sum_{s=1}^{K} \frac{s}{K} y^n_s. \tag{6}$$

Notice that the expected rate and the amount of storage used are a convex and a linear function of the placement parameters, respectively.

III. Formulation of RMSC in terms of the placement parameters and characterization of its dual problem

A. Formulation of RMSC

Using (3)-(6), the problem of finding the storage parameters that minimize the expected delivery rate

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIT.2019.2946361, IEEE Transactions on Information Theory

under the cache capacity constraint can be formulated as

$$\min_{\boldsymbol{y}} \quad \sum_{s=0}^{K-1} \frac{K-s}{s+1} \sum_{g\in\mathcal{G}_{s+1}} \pi^g_{s+1} \max_{n\in g} y^n_s \tag{7a}$$

$$\text{s.t.} \quad \sum_{n=1}^{N}\sum_{s=1}^{K} \frac{s}{K} y^n_s \le M, \tag{7b}$$

$$\sum_{s=0}^{K} y^n_s = 1, \qquad n\in[N], \tag{7c}$$

$$y^n_s \ge 0, \qquad n\in[N],\ s=0,\dots,K. \tag{7d}$$

B. Duality Framework and Derivation of Joint Rate and Storage Minimization Problem

Optimization problem (7) is convex and Slater's condition holds for it as all inequality constraints are affine [22, Section 5.2.3]. Hence, with (7) as primal, the duality gap between the primal and the corresponding dual problem is zero [22, Section 5.2]. To derive the dual problem, we form the Lagrangian that accounts for the capacity constraint (7b) as

$$L(\boldsymbol{y},\gamma) = \sum_{s=0}^{K-1} \frac{K-s}{s+1} \sum_{g\in\mathcal{G}_{s+1}} \pi^g_{s+1} \max_{n\in g} y^n_s + \gamma\left(\sum_{n=1}^{N}\sum_{s=1}^{K} \frac{s}{K} y^n_s - M\right)$$

which results in the Lagrange dual function

$$\begin{aligned} g(\gamma) = \min_{\boldsymbol{y}} \quad & L(\boldsymbol{y},\gamma) \\ \text{s.t.} \quad & \sum_{s=0}^{K} y^n_s = 1,\ n\in[N], \\ & y^n_s \ge 0, \qquad n\in[N],\ s=0,\dots,K. \end{aligned} \tag{8}$$

Then, the corresponding dual problem will be

$$\max_{\gamma\ge 0} \quad g(\gamma). \tag{9}$$

By dropping the terms that are independent of the placement parameters, (8) has the same minimizers as

$$\min_{\boldsymbol{y}} \quad \sum_{s=0}^{K-1} \frac{K-s}{s+1} \sum_{g\in\mathcal{G}_{s+1}} \pi^g_{s+1} \max_{n\in g} y^n_s + \gamma \sum_{n=1}^{N}\sum_{s=1}^{K} \frac{s}{K} y^n_s \tag{10a}$$

$$\begin{aligned} \text{s.t.} \quad & \sum_{s=0}^{K} y^n_s = 1,\ n\in[N], \\ & y^n_s \ge 0, \qquad n\in[N],\ s=0,\dots,K. \end{aligned} \tag{10b}$$

We call (10) the Joint Rate and Storage Minimization (JRSM) problem, as the objective is to minimize the total bandwidth (expected delivery rate) and storage cost of coded caching. Following the standard interpretation of the Lagrange multipliers, parameter γ can be viewed as the relative cost of storage per file. Moreover, since strong duality holds, for each storage capacity M, the optimal dual variable $\gamma^*(M)$ determines a pricing of the storage for which there exists the same minimizer to both the RMSC problem (7) and the Lagrangian minimization problem in (8) (or equivalently the JRSM problem in (10)). As a result, we derive the optimal solution of JRSM in Section IV as an intermediate step in solving RMSC.

IV. Optimal Solution to JRSM

Finding an analytical solution to (10) is challenging because of the presence of the $\max$ functions that operate over overlapping sets of parameters in the objective. These parameters are tied together by constraints (10b) for different values of s. The interplay between the outputs of the $\max$ function applied to the overlapping groups under constraints (10b) makes the analysis difficult. To facilitate the analysis, we establish a connection between the nonlinear part of (10a) and submodular set functions. This allows us to benefit from the results in submodular function analysis to find structures in the optimal solution to JRSM. Appendix B provides a review of submodular functions and the results relevant to our analysis in this paper.

A. An Equivalent Formulation of JRSM

The placement parameters corresponding to $s=0$ are $\{y^n_0\}_{n\in[N]}$, which determine the portion of bits that are not stored in any cache for each file n. Also, each set $g\in\mathcal{G}_1$ includes exactly one file, say $g=\{i\}$. Hence, $\max_{n\in g} y^n_0 = y^i_0$ and $\pi^g_0 = p_i$. Thus, the objective function (10a) can be written as

$$\sum_{n=1}^{N} K p_n y^n_0 + \sum_{s=1}^{K-1}\left[\frac{K-s}{s+1}\sum_{g\in\mathcal{G}_{s+1}} \pi^g_{s+1}\max_{n\in g} y^n_s + \frac{s}{K}\gamma\sum_{n=1}^{N} y^n_s\right] + \gamma\sum_{n=1}^{N} y^n_K.$$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIT.2019.2946361, IEEE Transactions on Information Theory

Notice that the first and last sums are in terms of parameters y_0^n and y_K^n, respectively, while the summation in the middle accounts for parameters y_s^n for $s \in [K-1]$.

Lemma 1: At optimality, $\sum_{n=1}^{N} Kp_n y_0^n + \gamma \sum_{n=1}^{N} y_K^n$ can be written as $\sum_{n=1}^{N}(Kp_n\alpha_n + \gamma(1-\alpha_n))z^n$ where $z^n = y_0^n + y_K^n$, and $\alpha_n = 1$ if $Kp_n < \gamma$ and $\alpha_n = 0$ if $Kp_n \geq \gamma$.

Proof: For a fixed value of z^n, we have $\sum_{n=1}^{N} Kp_n y_0^n + \gamma \sum_{n=1}^{N} y_K^n = \gamma \sum_{n=1}^{N} z^n + \sum_{n=1}^{N}(Kp_n - \gamma)y_0^n$. Hence, if $Kp_n < \gamma$, setting $y_0^n = z^n$ and $y_K^n = 0$ leads to the smallest objective and if $Kp_n \geq \gamma$, the smallest objective results for $y_0^n = 0$ and $y_K^n = z^n$. ∎

Corollary 1: For some $m \in \{0, \ldots, N\}$, we have $\alpha_n = 1, n > m$ and $\alpha_n = 0, n \leq m$.

Using Lemma 1, and the fact that $z^n = 1 - \sum_{s=1}^{K-1} y_s^n$, we get

$$\min_{\tilde{\boldsymbol{y}}, \boldsymbol{\alpha}} \quad \sum_{s=1}^{K-1} \frac{K-s}{s+1} \sum_{g \in \mathcal{G}_{s+1}} \pi_s^g \max_{n \in g} y_s^n \tag{11a}$$

$$+ \sum_{n=1}^{N} \sum_{s=1}^{K-1} \Big[\Big(\frac{s}{K} - 1 + \alpha_n \Big) \gamma - Kp_n \alpha_n \Big] y_s^n + l(\boldsymbol{\alpha})$$

$$\text{s.t.} \quad \sum_{s=1}^{K-1} y_s^n \leq 1, \, n \in [N], \tag{11b}$$

$$y_s^n \geq 0, \qquad n \in [N], \, s \in [K-1], , \tag{11c}$$

$$\alpha_n \in \{0, 1\}, \, n \in [N], \tag{11d}$$

as a problem equivalent to (10), where $\tilde{\boldsymbol{y}}$ is the same as $\boldsymbol{y}$, except for parameters y_0^n and y_K^n that are removed, and $l(\boldsymbol{\alpha}) = K \sum_{n=1}^{N} \alpha_n p_n + \gamma \sum_{n=1}^{N} (1 - \alpha_n)$.

To find structures in the optimal vector $\tilde{\boldsymbol{y}}$, assume that the optimal parameters α_n^* are known. Then, the optimization problem for $\tilde{\boldsymbol{y}}$ becomes

$$\min_{\tilde{\boldsymbol{y}}, t} \quad t + \sum_{n=1}^{N} \sum_{s=1}^{K-1} \Big[\Big(\frac{s}{K} - 1 + \alpha_n^* \Big) \gamma - Kp_n \alpha_n^* \Big] y_s^n \tag{12a}$$

$$\text{s.t.} \quad \sum_{s=1}^{K-1} \frac{K-s}{s+1} \sum_{g \in \mathcal{G}_{s+1}} \pi_s^g \max_{n \in g} |y_s^n| \leq t, \tag{12b}$$

$$\sum_{s=1}^{K-1} y_s^n \leq 1, \qquad n \in [N], \tag{12c}$$

$$y_s^n \geq 0, \qquad n \in [N], \, s \in [K-1]. \tag{12d}$$

In constraint (12b), we used $\max_{n \in g} |y_s^n|$, which is the l_∞-norm instead of $\max_{n \in g} y_s^n$. This does not affect the optimal solution as the two functions are equivalent in the nonnegative orthant specified by (12d) but makes the LHS in form of the l_∞-norm in Proposition B.1 of Appendix B.

Notice that objective function (12a) is linear, and both the objective function and the constraints are in terms of parameters y_s^n for $s \in [K-1]$, $n \in [N]$. For a linear objective function, if the feasible set is convex and bounded with a finite number of extreme points, then there exists an extreme point that is optimal [23, Section 2.5]. In the following, we will show that the feasible set defined by (12b)-(12d) satisfies these properties for any given value of t, and in particular for t^* at optimality, and derive structures in its extreme points. Any such structure readily implies a structure in at least one optimal solution to (12).

B. Connection to Submodular Functions

To find the extreme points of the region characterized by (12b), we establish a link to submodular functions. Let us define function

$$f_{\mathrm{c}}(\tilde{\boldsymbol{y}}) \triangleq \sum_{s=1}^{K-1} \frac{K-s}{s+1} \sum_{g \in \mathcal{G}_{s+1}} \pi_s^g \max_{n \in g} |y_s^n|.$$

The subscript c is used to highlight that this function returns the average rate due to the delivery of the bits that are cached in at least one of the caches in the system. We show that $f_{\mathrm{c}}(\tilde{\boldsymbol{y}})$ is the Lovász extension of a submodular set function. For that, consider the set $[(K-1)N]$. For each $s \in [K-1]$, objects $(s-1)N+1, \ldots, sN$ correspond to files $1, \ldots, N$, respectively. Notice that these objects belong to $[N]_{(s-1)N}$.

To define the corresponding set function, let us introduce the following for any $s \in [K-1]$ and $g \in \mathcal{G}_{s+1}$:

- Operator $u(s, g)$ that gives the set $\tilde{g} = \{(s-1)N + n \mid n \in g\}$ as output. This is basically a mapping from the files in g and set sizes s to the objects in $[(K-1)N]$. Notice that any resulting set $\tilde{g}$ is a subset of $[N]_{(s-1)N}$ for exactly one s.
- Sets $\tilde{\mathcal{G}}_{s+1} = \{u(s, g) \mid g \in \mathcal{G}_{s+1}\}$ and $\tilde{\mathcal{G}} = \bigcup_{s \in [K-1]} \tilde{\mathcal{G}}_{s+1}$.
- The inverse operators $s^{-1}(\tilde{g})$ and $g^{-1}(\tilde{g})$ that for $\tilde{g} \in \tilde{\mathcal{G}}$ return the unique s that satisfies $\tilde{g} \subset [N]_{(s-1)N}$, and the set $g = \{n \mid s^{-1}(\tilde{g})N + n \in \tilde{g}\}$, respectively.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIT.2019.2946361, IEEE Transactions on Information Theory

- Weights

$$\eta_{\tilde{g}} = \frac{K - s^{-1}(\tilde{g})}{s^{-1}(\tilde{g}) + 1} \pi_{s^{-1}(\tilde{g})}^{g^{-1}(\tilde{g})} \tag{13}$$

for all $\tilde{g} \in \tilde{\mathcal{G}}$. Notice that when $|\tilde{g}| = 1$, $g^{-1}(\tilde{g}) = \{i\}$ which is a singleton. In that case, $\pi_{s^{-1}(\tilde{g})}^{g^{-1}(\tilde{g})} = p_i$.

Using the operators and parameters defined above, $f_{\rm c}(\tilde{\boldsymbol{y}})$ can be written as

$$f_{\rm c}(\tilde{\boldsymbol{y}}) = \sum_{\tilde{g} \in \tilde{\mathcal{G}}} \eta_{\tilde{g}} \max_{n \in g^{-1}(\tilde{g})} |y^n_{s^{-1}(\tilde{g})}|. \tag{14}$$

Notice that (14) has the form of the norm function in Proposition B.1 and as a direct consequence we have the following proposition:

Proposition 2: Function $f_{\rm c}(\tilde{\boldsymbol{y}})$ is a norm and is the Lovász extension of the submodular function

$$F_{\rm c}(A) = \sum_{\tilde{g} \in \tilde{\mathcal{G}}: A \cap \tilde{g} \neq \emptyset} \eta_{\tilde{g}}, \tag{15}$$

where $A \subset [(K-1)N]$.

From Proposition 2, one concludes that constraint (12b) characterizes a norm-ball of radius t.

For $A \subset [N]_{(s-1)N}$, let us define $P(A) = \sum_{n \in g^{-1}(A)} p_n$. Then, for the extreme points of the norm-ball, we have the following lemma.

Lemma 2: The extreme points of the norm-ball $f_{\rm c} \leq t$ are of the form

$$\frac{t}{\frac{K-s}{s+1}\left[1 - (1 - P(A))^{s+1}\right]} \boldsymbol{v}, \tag{16}$$

where vector $\boldsymbol{v} \in \{-1, 0, 1\}^{KN}$, $\text{Supp}(\boldsymbol{v}) = A$, and set A is a subset of $[N]_{(s-1)N}$ for an $s \in [K-1]$.

Proof: Based on Proposition B.2, the extreme points of the unit ball $f_{\rm c} \leq 1$ are closely connected to the set of stable inseparable subsets of $[(K-1)N]$ with regard to $F_{\rm c}$. We first argue that all subsets of $[(K-1)N]$ are stable. Consider a set $A \subset [(K-1)N]$. Augment A with a new object i to get $A \cup \{i\}$. Without loss of generality, let $s^{-1}(\{i\}) = \hat{s}$. Since $\tilde{g} = \{i\}$ belongs to $\mathcal{G}_{\hat{s}+1}$ with $\eta_{\tilde{g}} > 0$ and it does not intersect with A, we have $f_{\rm c}(A \cup \{i\}) > F_{\rm c}(A)$. Hence, any set $A \subset [(K-1)N]$ is stable with respect to $F_{\rm c}$. Consequently, every subset of $[N]_{(s-1)N}$ for $s \in [K-1]$ is also stable.

To find the inseparable sets, consider $A \subset [(K-1)N]$. Let $B_s = \{i \in A \mid s^{-1}(\{i\}) = s\}$. A necessary condition for A to be inseparable is to have only one nonempty B_s. To show this, partition A to subsets $B_s, s \in [K-1]$. Notice that each group $\tilde{g} \subset \tilde{\mathcal{G}}$ is a subset of exactly one $[N]_{(s-1)N}, s \in [K-1]$. Hence, if two or more subsets B_s are nonempty, then $F_{\rm c}(A) = \sum_{B_s \neq \emptyset} f_{\rm c}(B_s)$ and A is separable. Now, consider the case where only one B_s, say $B_{\hat{s}}$, is nonempty. In this case, $A \subset [N]_{(\hat{s}-1)N}$ and A can only have nonempty intersections with sets $\tilde{g} \in \tilde{\mathcal{G}}_{\hat{s}+1}$. Since $\hat{s} \geq 1$, for any partitioning of A to $P_1, \ldots, P_J$ for some J, there is at least one group $\tilde{g} \in \tilde{\mathcal{G}}_{\hat{s}+1}$ with $|\tilde{g}| \geq 2$ that intersects with both P_i and P_j for every pair $i \neq j$. Hence, $F_{\rm c}(A) < \sum_{i=1}^{J} f_{\rm c}(P_i)$. As a result, the set of all stable inseparable subsets of $[(K-1)N]$ with regard to $F_{\rm c}$ is $\mathcal{A} = \{A \mid A \subset [N]_{(s-1)N}, s \in [K-1]\}$.

According to Proposition B.2, the support of every extreme point of the norm-ball of $f_{\rm c}$ belongs to $\mathcal{A}$. Further, the nonzero entries of the extreme point vector that corresponds to $A \in \mathcal{A}$ is either of $\pm 1/F_{\rm c}(A)$. Using Proposition 2:

$$\begin{aligned} F_{\rm c}(A) &= \sum_{\tilde{g} \subset \tilde{\mathcal{G}}: A \cap \tilde{g} \neq \emptyset} \eta_{\tilde{g}} = \sum_{\substack{\tilde{g} \subset \tilde{\mathcal{G}}_{s^{-1}(A)}: \\ A \cap \tilde{g} \neq \emptyset}} \eta_{\tilde{g}} \\ &= \frac{K - s^{-1}(A)}{s^{-1}(A) + 1} \left(1 - (1 - P(A))^{s^{-1}(A)+1}\right) \end{aligned} \tag{17}$$

where we used the facts that 1) for $A \in \mathcal{A}$, all entries of A belong to only one $[N]_{(s-1)N}$, so $s^{-1}(A)$ and $g^{-1}(A)$ are well defined, 2) $\eta_{\tilde{g}} = \frac{K - s^{-1}(\tilde{g})}{s^{-1}(\tilde{g})+1} \pi_{s^{-1}(\tilde{g})}^{g^{-1}(\tilde{g})}$ and 3) $\sum_{\substack{\tilde{g} \subset \tilde{\mathcal{G}}_{s^{-1}(A)}: \\ A \cap \tilde{g} \neq \emptyset}} \eta_{\tilde{g}}$ equals the probability of having a demand vector with at most $s^{-1}(A) + 1$ distinct files from $[N]$ that has at least one file in $g^{-1}(A)$. The use of $\mathcal{A}$ and (17) in Proposition B.2 and scaling the radius of the ball from 1 to t results in (16). ∎

Corollary 2: For an extreme point $\tilde{\boldsymbol{y}}$ of the norm-ball defined by (12b), for all $y^n_s > 0$, we have $s = \hat{s}$, for exactly one $\hat{s} \in [K-1]$.

Theorem 1: There is an optimal solution to the JRSM problem in (10) which is of form

$$(y^n_s)^* = \begin{cases} 1, & s = 0, n \in [N - n^*]_{n^*}, \\ 1, & s = s^*, n \in [n^*], \\ 0, & \text{otherwise}, \end{cases} \tag{18}$$

for some $s^* \in [K]$ and some $n^* \in \{0, \ldots, N\}$.[4]

Proof: See Appendix C. ∎

[4]Notice that $n^* = 0$ corresponds to the case where for all $n \in [N]$: $y^n_0 = 1$ and $y^n_s = 0, s > 0$.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIT.2019.2946361, IEEE Transactions on Information Theory

Theorem 1 implies that for every $\gamma \geq 0$ there exists an optimal solution to the JRSM problem that is integral. For better illustration, we write such an optimal vector $\boldsymbol{y}$ in the matrix form Y as follows. Matrix Y has N rows corresponding to the files and $K+1$ columns corresponding to the cardinality of subsets of caches. In particular, $Y_{n,s} = y_s^n, n \in [N], s = 0, \ldots, K$. Based on the structures found for y_s^n in Theorem 1, Y is of form

$$Y = \begin{array}{c} \\ 1 \\ 2 \\ \vdots \\ n^*-1 \\ n^* \\ n^*+1 \\ \vdots \\ N \end{array} \begin{array}{c} \begin{array}{cccccccc} 0 & 1 & \ldots & s^*-1 & s^* & s^*+1 & \ldots & K \end{array} \\ \left[\begin{array}{cccccccc} \mathbf{0} & 0 & \ldots & 0 & \mathbf{1} & 0 & \ldots & 0 \\ \mathbf{0} & 0 & \ldots & 0 & \mathbf{1} & 0 & \ldots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & 0 & \ldots & 0 & \mathbf{1} & 0 & \ldots & 0 \\ \mathbf{0} & 0 & \ldots & 0 & \mathbf{1} & 0 & \ldots & 0 \\ \mathbf{1} & 0 & \ldots & 0 & \mathbf{0} & 0 & \ldots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ \mathbf{1} & 0 & \ldots & 0 & \mathbf{0} & 0 & \ldots & 0 \end{array}\right] \end{array}, \quad (19)$$

i.e., i) all entries are either 0 or 1, ii) each row has exactly one entry 1, iii) at most one column with index $s \geq 1$ has nonzero entries, iv) for that column all entries are 1 for rows $1, \ldots, n^*$ and 0 for rows $n^*+1, \ldots, N$, for some $n^* \in [N]$. As a result, for all values of $\gamma \geq 0$ there is an optimal solution with matrix form (19). Hence, we have the following corollary:

Corollary 3: There exists a finite set $\mathcal{Y}^*$ of vectors that correspond to matrices of form (19) where $|\mathcal{Y}^*| \leq KN+1$ and that set includes at least one optimal solution to the JRSM problem (10) for every $\gamma \geq 0$.[5]

The structure of the optimal solution to JRSM in Theorem 1 has direct implications about the solution of the RMSC problem. In particular, based on the duality framework detailed in Section III-B, for $\gamma = \gamma^*$, i.e., the optimal Lagrange multiplier that solves the dual problem (9), if the optimal solution to JRSM with the structure in Theorem 1 uses a storage equal to M in the capacity constraint of RMSC, this solution is also optimal to the RMSC problem. This implies that for certain storage capacities in the RMSC problem, its optimal solution has the same structure as in Theorem 1. In Section V, we fully investigate the solution to the RMSC problem for the general storage capacity M and its connection to the solution of the JRSM problem.

[5]We show in Appendix E that for specific values of γ, there are infinite number of solutions to the JRSM problem.

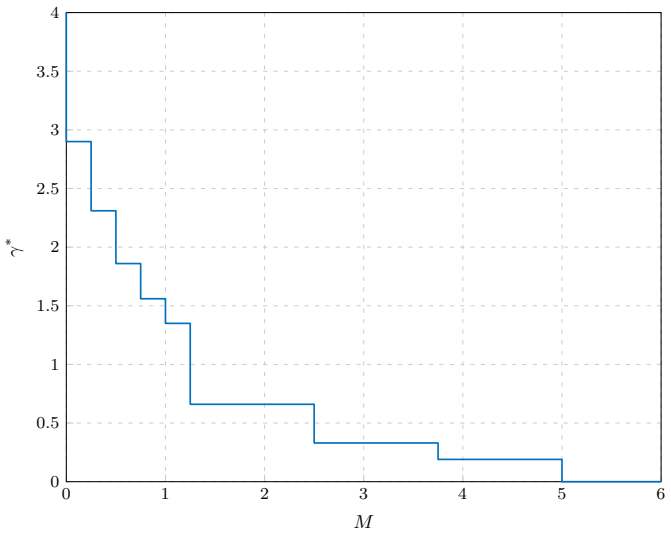


Fig. 2. Here, $N = 5$, $K = 4$, and the popularity distribution is Zipf with parameter $\alpha = 1$.[1]

V. Optimal Solution to RMSC

A. Optimal Solution of RMSC in Terms of Optimal JRSM Solution

Assuming that the optimal dual parameter γ^* is known, we derived structures in the minimizers of the Lagrangian or equivalently in the optimal solution of JRSM. The derived structures limited the search space for the optimal JRSM solution to $KN+1$ vectors specified by Theorem 1. In this section, we derive the optimal solution to RMSC by building on the results we derived for the solution of JRSM in Theorem 1. For that, let us define two sets as follows:

Definition 1: Sets $\mathcal{M}$ and $\mathcal{R}$ are finite sets defined by storage values $\{m(\boldsymbol{y}) \mid \boldsymbol{y} \in \mathcal{Y}^*\}$ and expected rates $\{r(\boldsymbol{y}) \mid \boldsymbol{y} \in \mathcal{Y}^*\}$, respectively.

To characterize the solution of RMSC, we take the following two steps. First, we assume that set $\mathcal{Y}^*$ and consequently set $\mathcal{M}$ are known. Based on this assumption, we derive the optimal dual parameter γ^* as a function of storage capacity M in the primal problem. Second, we use the derived γ^*-M relationship to find set $\mathcal{Y}^*$ and derive the optimal solution to RMSC.

Lemma 3: The optimal dual parameter γ^* and the storage capacity M in the primal RMSC problem satisfy the following:

1) Parameter γ^* is non-increasing in M;
2) For certain storage capacities M, a continuum of dual parameters γ^* are optimal;

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIT.2019.2946361, IEEE Transactions on Information Theory

3) For every two consecutive values $M_1, M_2 \in \mathcal{M}, M_1 < M_2$, any $M \in [M_1, M_2]$ leads to the same dual optimal parameter γ^*.

Proof: See Appendix D. ∎

Lemma 3 implies a stairwise relationship between the optimal dual parameter γ^* and the storage capacity M in the primal problem. An illustration of this relationship is shown in Fig. 2. The fact that γ^* is non-increasing in M is in agreement with the interpretation of the Lagrange multiplier γ^* as the (relative) price per unit of storage [22]: as more storage becomes available, the storage price remains the same or decreases. The second point in the lemma corresponds to the vertical line segments in Fig. 2. Based on Definition 1, set $\mathcal{M}$ which is derived from $\mathcal{Y}^*$ is finite and has at most $KN+1$ members. However, γ is a continuous variable and for every $\gamma \geq 0$ there is an optimal solution to JRSM in $\mathcal{Y}^*$. Hence, an interval of γ values must map to the same M. Third, a range of values of M are mapped into the same γ^*. Notice that parameter M in the primal problem can take any nonnegative value, while $\mathcal{M}$ is a set of discrete values and is of finite size. Since every $M \geq 0$ corresponds to at least one optimal dual parameter γ, then a continuum of values for M must map to the same γ^*. We show that parameter γ^* and the two solutions $\boldsymbol{y}_1, \boldsymbol{y}_2 \in \mathcal{Y}^*$ that lead to the two endpoints $(m(\boldsymbol{y}_1), \gamma^*)$ and $(m(\boldsymbol{y}_2), \gamma^*)$ of the line segment are related by $\gamma^* = \frac{r(\boldsymbol{y}_1)-r(\boldsymbol{y}_2)}{m(\boldsymbol{y}_2)-m(\boldsymbol{y}_1)}$. Notice that $m(\boldsymbol{y}_1), m(\boldsymbol{y}_2) \in \mathcal{M}$ and $r(\boldsymbol{y}_1), r(\boldsymbol{y}_2) \in \mathcal{R}$. In particular, if we sort members of $\mathcal{M}$ in increasing order as $0 = M_0 < M_1 < M_2 < \ldots < M_l = N$, then rates R_i that correspond to storage values M_i follow ordering $K = R_0 > R_1 > \ldots > R_l = 0$. Hence

$$\gamma^*(M) = \begin{cases} [\frac{R_i - R_{i+1}}{M_{i+1} - M_i}, \frac{R_{i-1} - R_i}{M_i - M_{i-1}}], & M = M_i \\ \frac{R_{i-1} - R_i}{M_i - M_{i-1}}, & M_{i-1} < M < M_i \end{cases}$$

with $\gamma^*(M_0 = 0) = [\frac{K - R_1}{M_1}, +\infty]$ and $\gamma^*(M_l = N) = [0, \frac{R_{l-1}}{N - M_{l-1}}]$.

The next theorem determines the relationship between the optimal solution of RMSC and the optimal solution of JRSM which was derived in Theorem 1.

Theorem 2: The RMSC problem (7) has an optimal solution

$$y^*_{\text{RMSC}}(M) = \begin{cases} y^*_{\text{JRSM}}(M), & M \in \mathcal{M} \\ \frac{M_{\text{u}} - M}{M_{\text{u}} - M_{\text{l}}} y^*_{\text{JRSM}}(M_{\text{l}}) + \frac{M - M_{\text{l}}}{M_{\text{u}} - M_{\text{l}}} y^*_{\text{JRSM}}(M_{\text{u}}), & M \notin \mathcal{M} \end{cases} \quad (20)$$

where $y^*_{\text{JRSM}}(m)$ is the optimal solution of JRSM of the form in Theorem 1 that uses storage m, and M_{l} and M_{u} are the largest element smaller than M and smallest element larger than M in $\mathcal{M}$, respectively.

Proof: See Appendix E ∎

Optimality of Memory Sharing: Theorem 2 essentially extends a result known for the optimal solution of RMSC for uniform demands to the general case where demands can be nonuniform. To elaborate, it has been shown that for uniform demands, if $M \in \{1, \frac{N}{K}, 2\frac{N}{K}, \ldots, K\frac{N}{K}\}$, then the optimal solution of RMSC is in the form in (19) for some $s^* \in [K]$ and $n^* = N$ [15]–[17]. In particular, for uniform demands $\mathcal{M}_{\text{uniform}} = \{0, \frac{N}{K}, 2\frac{N}{K}, \ldots, K\frac{N}{K}\}$[6]. For other values of M, the optimal solution could be obtained by memory sharing between the two values of storage in $\mathcal{M}_{\text{uniform}}$ closest to M. Theorem 2 shows that the same result is valid for nonuniform demands except for the fact that n^* might be any value between 0 and N, depending on the popularity distribution of files. As a result, we propose the following terminology:

Definition 2: For a given number of caches and popularity distribution of files, we call set $\mathcal{M}$ the set of base-cases of the RMSC problem.

Based on Theorem 1, for base-cases of RMSC, there exists an optimal solution which is integral. Also, from Theorem 2, for other storage capacities, the optimal solution to RMSC can be obtained by memory sharing between the solutions of two certain base-cases.

B. Algorithm to Derive $\mathcal{M}$

We derive an algorithm with a worst-case complexity of $O(K^2N^2)$ to find set $\mathcal{Y}^*$ and consequently $\mathcal{M}$ for any given number of caches and popularity distribution of files. With $\mathcal{M}$ determined, Theorem 2 analytically solves the RMSC problem for any cache capacity.

To find $\mathcal{Y}^*$, we need to search over the $KN+1$ possibilities for $\boldsymbol{y}^*$ of form (18). For each such vector $\boldsymbol{y}$, the storage it uses can be written as a convex combination of the storage used by two other vectors $\boldsymbol{y}_1$ and $\boldsymbol{y}_2$ that satisfy $m(\boldsymbol{y}_1) \leq m(\boldsymbol{y})$ and $m(\boldsymbol{y}_2) \geq m(\boldsymbol{y})$. In other words, $m(\boldsymbol{y}) = m(\theta \boldsymbol{y}_1 + (1-\theta)\boldsymbol{y}_2), 0 \leq \theta \leq 1$.[7] If for such $\boldsymbol{y}_1, \boldsymbol{y}_2$, we further have $r(\boldsymbol{y}) \geq \tilde{m}(\boldsymbol{y})$, then $\boldsymbol{y}$ does not belong to $\mathcal{Y}^*$. Hence, by removing such vectors form the $KN+1$ possibilities, the remaining set

[6] In the case of $M = 0$, we have $n^* = 0$ for the optimal RMSC solution.

[7] except for the two vectors with $m(\boldsymbol{y}) \in \{0, N\}$.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIT.2019.2946361, IEEE Transactions on Information Theory

Algorithm 2 Procedure to Determine the Set of Base-Cases $\mathcal{M}$

procedure BASE($K, N, \{p_n\}$)
  # Calculate Storage and Rate for the $KN+1$ matrices of form (19)
  $Y_0 \leftarrow \mathbf{0}_{N\times(K+1)}$, $Y_0(1:N,0) \leftarrow 1$, $M_0 \leftarrow 0$, $R_0 \leftarrow K$
  for $s = 1, \ldots, K$ **do**
    for $n = 1 : N$ **do**
      $i \leftarrow (s-1)N + n$
      $Y_i \leftarrow \mathbf{0}_{N\times(K+1)}$, $Y_0(n+1:N,0) \leftarrow 1$, $Y_i(1:n,s) \leftarrow 1$
      $M_i \leftarrow m(Y)$
      $R_i \leftarrow r(Y)$
    end for
  end for
  $(M, R, Y) \leftarrow \text{sort}_M(M, R, Y)$ # relabel (M_i, R_i, Y_i) tuples in increasing order of M_i
  # Build $\mathcal{M}$, $\mathcal{R}$ and $\mathcal{Y}$ by keeping solutions that outperform memory sharing between other cases
  $(\mathcal{Y}_0, \mathcal{M}_0, \mathcal{R}_0) \leftarrow (\mathbf{0}_{N\times(K+1)}, 0, K)$
  $c \leftarrow 0$
  for $i = 1, \ldots, NK+1$ **do**
    for $j = i+1 : NK+1$ **do**
      $R_{\text{msh}} \leftarrow \frac{M_j - M_i}{M_j - \mathcal{M}_c}\mathcal{R}_c + \frac{M_i - \mathcal{M}_c}{M_j - \mathcal{M}_c}R_j$
      if $R_i < R_{\text{msh}}$ **then**
        $c \leftarrow c+1$
        $(\mathcal{Y}_c, \mathcal{M}_c, \mathcal{R}_c) \leftarrow (Y_i, M_i, R_i)$
        break
      end if
    end for
  end for
end procedure

of vectors constitutes $\mathcal{Y}^*$. The BASE procedure in Algorithm 2 implements this process by starting from Y with all entries in the first column equal to 1. This correspond to storage 0 and rate K and belongs to $\mathcal{Y}^*$. It then proceeds to the next $\boldsymbol{y}$ with the smallest storage value. It checks whether it outperforms memory sharing between the $\boldsymbol{y}$ that is already in $\mathcal{Y}^*$ with the largest storage and every remaining vector that uses more storage compared to $\boldsymbol{y}$. If that is the case, it adds the new vector to $\mathcal{Y}^*$, otherwise drops the vector and proceeds to the next vector.

Fig. 3 shows the expected delivery rate of the proposed method versus the cache capacity for a nonuniform distribution of files that follows a Zipf density with parameter 1.4. The expected rate is once calculated based on the solution obtained by Algorithm 2 and once by solving RMSC numerically. We observe that the resulting optimal rates are in complete agreement. Fig. 3 also shows the amount of storage used to cache subsets of files that are exclusively stored in subsets of caches with different cardinalities $s \in [K]$. In other words, for each s, it shows $Q_s \triangleq \sum_{n=1}^{N} \frac{s}{K} y_s^n$ as a function of the cache capacity. As we expect from our analysis, either one or two values of Q_s can be positive for each choice of M.

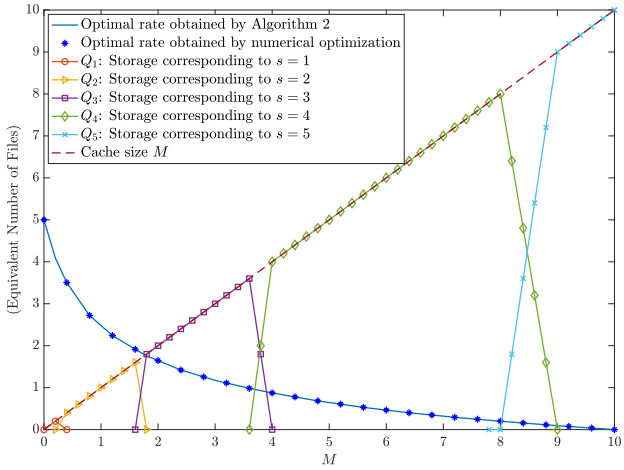


Fig. 3. The effect of cache size on expected delivery rate and the amount of storage used to cache subsets of files that are exclusively stored in subsets of caches with cardinalities $1, \ldots, K$ for $K = 5$, $N = 10$. Here, the popularity of files follows a Zipf distribution with parameter 1.4.

C. Numerical Exploration

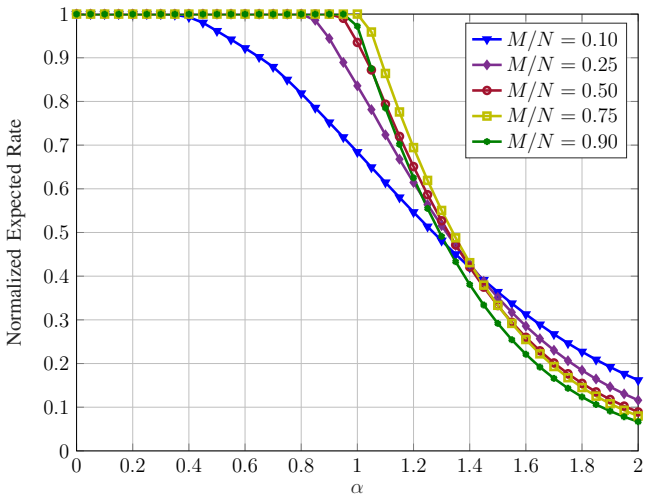
Fig. 4 shows the joint effect of the nonuniformity of the file request probabilities and the cache size M. The nonuniformity of the probability mass function is controlled by parameter α of Zipf distribution. Fig. 4a shows the expected rate of RMSC for $0 \leq \alpha \leq 2$ normalized by the expected rate of SCC for the corresponding value of α when the the placement for uniform demands is used. This normalization puts the curves for different cache sizes in the same scale for better visual clarity and more significantly makes the curves more interpretable. In particular, we observe that for a fixed cache size, the normalized expected rate remains almost

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIT.2019.2946361, IEEE Transactions on Information Theory

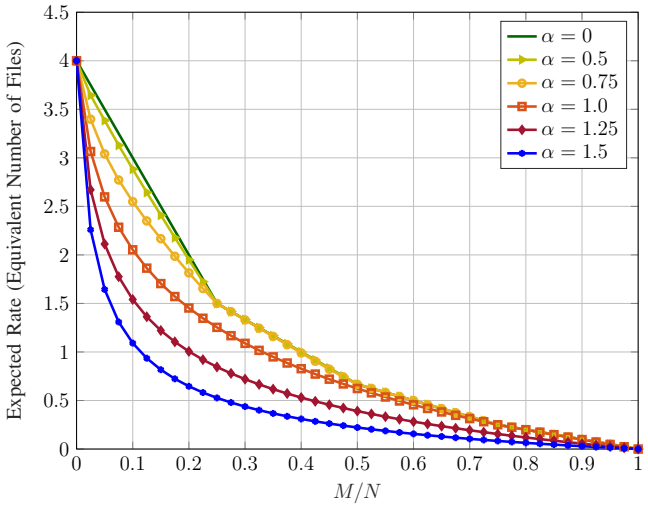
equal to 1 as α is increased from 0 up to some threshold value. In other words, the optimal the delivery rate for a slightly nonuniform popularity distribution is almost identical to the delivery rate given by the placement that treats the files as if they were uniformly popular. The threshold value of α depends on the available cache capacity. In general, as the Zipf distribution gets more heavy-tailed (smaller α) the normalized rate gets closer to 1. Fig. 4b shows the expected delivery rate of RMSC versus cache size for different values of α. Consistent with our previous observation, for heavy-tailed popularity distributions, like the cases with $\alpha = 0, 0.5, 0.75$, the optimal delivery rates are close to each other and are only different when cache storage is scarce ($M/N \leq 0.25$).

Comparison to the caching scheme of [14]: Reference [14] also proposed the use of a two-group or three-group strategy for the caching of files with nonuniform demands. Fig. 5 compares the delivery rates obtained by RMSC to those obtained by [14]. Accordingly, Table II provides the groupings of the files made by the two techniques, where n^* and N_1 represent the index of the last popular files given by RMSC and [14], respectively.[8] Also, R^* and R_1 represent the respective expected delivery rates of the two techniques. The superior performance of RMSC is evident in both Fig. 5 and Table II. In Table II, one observes that the set of popular files given by RMSC and [14] are identical for specific set of problem parameters but the delivery rates are different. This is because despite the identical grouping of files, the placement of the popular files and the delivery algorithms are still different for the two techniques. In particular, [14] follows a randomized placement algorithm while RMSC uses a deterministic placement technique. The randomized algorithm simplifies the placement at the expense of a higher delivery rate.

Comparison to the information theoretic outer bound: Also plotted in Fig. 5 are two lower-bounds on the expected delivery rate of nonuniform demands which are reported in [14, Theorem 1] and [18, Theorem 2]. The lower-bound in [18, Theorem 2] is over the caching schemes with uncoded placement of content. Since the SCC algorithm relies on uncoded placement of content, the bound in [18, Theorem 2] must hold for the expected delivery rate of RMSC. The gap between the lower-bound in [18, Theorem 2] and the expected rate of RMSC is small in general, suggesting that the performance of the RMSC is close to the optimal performance for



(a) Normalized expected rate versus the Zipf parameter for different cache sizes

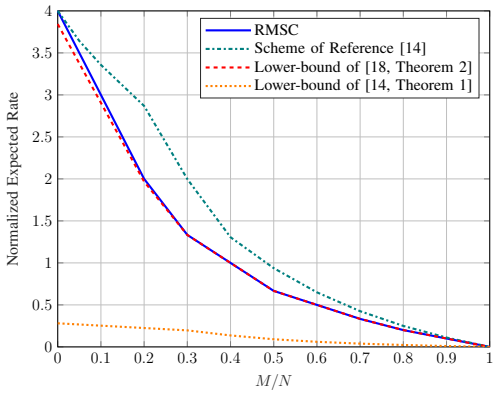


(b) Expected rate versus the normalized cache size for different Zipf parameters

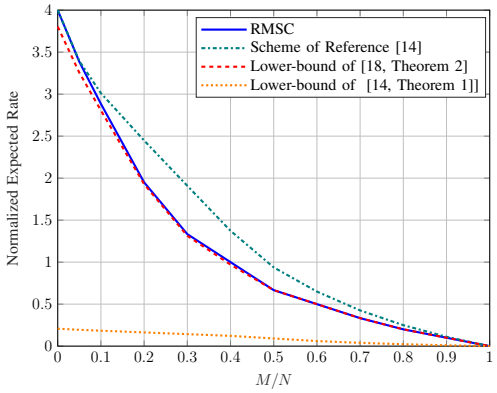
Fig. 4. Joint effect of cache size and nonuniformity of the file request probabilities on the expected delivery rate. In (a), the expected rates are normalized by the expected rate of the delivery algorithm SCC when the placement for uniform demands is used instead of the optimal placement. Here $K = 4$ and $N = 40$.

[8]For the caching technique in [14], it is possible for N_1 to be smaller than M. In Table II, this is the case for $M = 4$ ($M/N = 0.1$). In such cases, part of the memory would remain unused if only the popular files were to be cached. In such a scenario, the authors suggest that each cache stores the entirety of the first M files, and if M is not an integer, use the remaining $M - \lfloor M \rfloor$ capacity for the partial caching of file $\lfloor M \rfloor + 1$. In this scenario, uncoded messages are used to deliver the files that are not fully cached. For more details refer to [14, Section V].

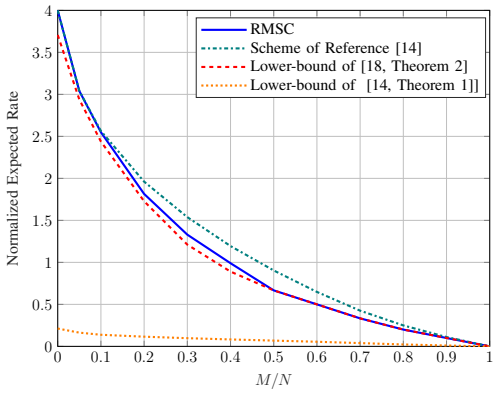
This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIT.2019.2946361, IEEE Transactions on Information Theory



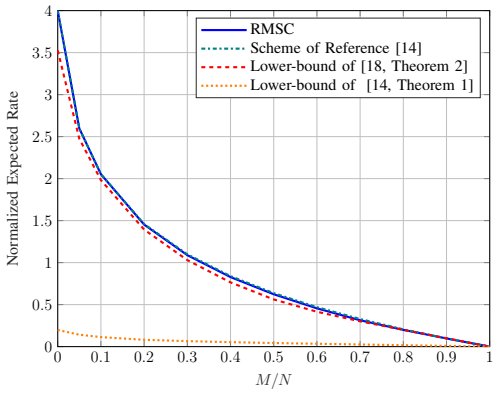
(a) $\alpha = 0.25$



(b) $\alpha = 0.5$



(c) $\alpha = 0.75$



(d) $\alpha = 1$

Fig. 5. Comparison of the delivery rates of RMSC and method of [14] with the information theoretic lower-bounds for different parameters α of Zipf distribution. In all cases, $N = 40$.

TABLE II. Comparison of sets of popular files of RMSC and [14] and the resulting rates. In all cases $N = 40$.

α	M	M/N	n^*	N_1	R_1/R^*
0.25	4	0.1	40	0	1.1192
	16	0.4	40	40	1.3056
	32	0.7	40	40	1.2753
0.5	4	0.1	16	2	1.0424
	16	0.4	40	32	1.3732
	32	0.7	40	40	1.2753
0.75	4	0.1	8	3	1.0059
	16	0.4	40	20	1.2063
	32	0.7	40	40	1.2753
1	4	0.1	4	3	1.000
	16	0.4	22	14	1.014
	32	0.7	38	26	1.047

coded caching with uncoded prefetching of content. More specifically, one can make the following two observations. First, for a fixed cache size, this gap increases as the file popularity distribution becomes more nonuniform (larger α). Second, for a fixed value of α, this gap shrinks and approaches zero as M/N increases. The existence of this performance gap is due to the sub-optimality of the SCC procedure as the delivery algorithm that we explore in detail in the discussion section that will follow.

For completeness of our numerical exploration, we also include the lower-bound derived in [14, Theorem 1] in Fig. 5. Unlike [18, Theorem 2], the bound in [14, Theorem 1] applies to all caching schemes regardless of whether coded or uncoded prefetching is used. However, we found this bound to be loose in general. This can be seen at the extreme case of $M/N = 0$, where the minimum amount of information that can be sent over the channel is equal to the number of distinct files requested. Yet, we see that the lower-bound in [14, Theorem 1] is considerably smaller than this value, suggesting that the bound is loose. Notice that the bound in [14, Theorem 1] is derived to prove order-optimality of the caching scheme proposed in [14] and is not guaranteed to be a tight bound on the optimal rate.

Discussion of the performance gap to the information-theoretic lower-bound: The existence of the gap between the expected rate of RMSC and the lower-bound in [18, Theorem 2], if not fully, is at least partially caused by the insensi-

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIT.2019.2946361, IEEE Transactions on Information Theory

tivity of the SCC algorithm in Algorithm 1 to the presence of duplicate requests in the demand vector. In other words, if multiple caches request identical files, it still delivers them as if the files were distinct. For instance, consider the case that $M/N = 0$ and all caches request the same file. In that case, no side information is available at the caches and all requests must be delivered by uncoded messages. The SCC algorithm delivers the requests by transmitting the requested file K times. This is clearly suboptimal, as in this case it suffices to transmit the single file requested by all caches only once. This inefficiency of the original SCC algorithm for redundant requests becomes more complex to characterize for $M > 0$, but it has been thoroughly investigated in the literature [24], [25]. In particular, a modified version of the SCC algorithm was proposed in [25, Section IV.B and Appendix C.A], which resolves this inefficiency and is shown to achieve the optimal memory-rate tradeoff for the case of uniform demands when uncoded placement is used [25]. More specifically, for any given demand vector $\boldsymbol{d}$, the modified SCC algorithm first chooses a set of leader caches $\mathcal{U}$ that have the property that they have all requested distinct files in $\boldsymbol{d}$. Hence, the number of leader caches is between 1 and K depending on $\boldsymbol{d}$. Then, contrary to Algorithm 1, which greedily transmits the binary sums $\oplus_{k\in\mathcal{S}} X^{d_k}_{\mathcal{S}\setminus k}$ for every $\mathcal{S} \subset [K]$, the modified algorithm transmits the binary sum for $\mathcal{S}$ only if $\mathcal{S} \cap \mathcal{U} \neq \emptyset$. The authors prove that with extra processing at non-leader caches, the messages transmitted by the modified SCC algorithm are enough for retrieving all the requested files at the caches. This difference between the original and modified SCC algorithms can explain the behavior of the performance gap in Fig. 5. In particular, as M/N increases, the portion of bits cached at subsets $\mathcal{S} \subset [K]$ with large $|\mathcal{S}|$ increases. At the same time, larger $|\mathcal{S}|$ implies a higher probability that $\mathcal{S}$ includes at least one leader cache. Hence, the amount of information transmitted similarly by the original and modified SCC algorithms increases as M/N increases. As a result, the sub-optimality of the SCC algorithm will have a minimal effect on the delivery rate and the gap of the rate of RMSC to the lower-bound shrinks as M/N increases.[9] Similarly, for a fixed M/N ratio, as α increases, more duplicate requests for the more popular files occur in the demand vector due to their considerably higher probability of request. Effectively, this reduces the number of leader caches for the different demand vectors on average. By reducing the number of subsets $\mathcal{S} : \mathcal{S} \cap \mathcal{U} \neq \emptyset$, this directly translates into the necessity of transmission of a smaller number of coded messages, and therefore, a larger gap between the performance of the original and the modified SCC algorithms. This analysis explains why the gap between RMSC and the lower-bound increases as α increases.

VI. Conclusion and Future Work

In this paper, we applied the structured clique cover delivery algorithm that was proposed for decentralized coded caching to deliver files with nonuniform request probabilities. We fully characterized the structure of the optimal placement parameters. We showed that for a finite set of cache capacities, called base-cases, the optimal placement follows the two group strategy that does not cache a subset of less popular files, but treats the other files selected for caching identically regardless of their popularities. A polynomial time procedure was also proposed to derive this set of cache capacities. We further showed that the optimal placement parameters for other storage capacities can be obtained by memory sharing between certain base cases. In this scenario, a grouping of files into two or three groups is optimal.

Motivated by our numerical results as well as the fact that for uniform demands, the modified SCC algorithm proposed in [25] results in the exact memory-rate tradeoff for caching with uncoded prefetching, it is worthwhile to explore whether an analysis similar to what we presented in this paper can characterize the optimal placement for coded caching with the modified SCC algorithm of [25] for delivery. Furthermore, it is of interest to see how the rate of such a caching scheme compares to the information-theoretic lower-bound on the expected delivery rate of caching with uncoded prefetching.

[9] A similar trend can be seen in [25, Fig. 5a] where the gap between the expected rates of the SCC algorithm and the (optimal) modified SCC algorithm vanishes as M/N increases, for the case of uniform demands.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIT.2019.2946361, IEEE Transactions on Information Theory

APPENDIX A
PROOF OF PROPOSITION 1

Proof: Assume that there exists an optimal placement $\mathcal{P}^*$ of files for which there exist distinct subsets $\mathcal{S}_1$ and $\mathcal{S}_2 : |\mathcal{S}_1| = |\mathcal{S}_2|$ such that $x^n_{\mathcal{S}_1} \neq x^n_{\mathcal{S}_2}$. Since the popularity of files is identical for the different caches, the delivery rate remains the same if we use any permutation $\text{perm}(k)$ of the cache labels in the placement parameters $\{x^n_{\mathcal{S}}\}_{\mathcal{S}\subset[K]}$. We denote the placement over the permuted cache labels by $\mathcal{P}^*_{\text{perm}}$. More specifically, we relabel cache k to $\text{perm}(k)$ for $k \in [K]$, and use the placement parameters over the relabeled caches. In particular, if under $\mathcal{P}^*$ we have $x^n_{\mathcal{S}}\big|_{\mathcal{P}^*} = c$, then under $\mathcal{P}^*_{\text{perm}}$, we set $x^n_{\text{perm}(\mathcal{S})}\big|_{\mathcal{P}^*_{\text{perm}}} = c$, where $\text{perm}(\mathcal{S}) = \{\text{perm}(k) \mid k \in \mathcal{S}\}$. There exists $K!$ permutations of K cache labels, leading to placements $\mathcal{P}^*_{\text{perm},i}, i = 1, \ldots, K!$, all with the optimal delivery rate R^*. Hence:

$$
\begin{aligned}
R^* &= \frac{1}{K!}\sum_{i=1}^{K!} r(\boldsymbol{x})\big|_{\mathcal{P}^*_{\text{perm},i}} \\
&= \frac{1}{K!}\sum_{i=1}^{K!}\sum_{\substack{\mathcal{S}:\mathcal{S}\subset\mathcal{K}\\ \mathcal{S}\neq\emptyset}} \mathbb{E}_{\boldsymbol{d}}\left(\max_{k\in\mathcal{S}} x^{d_k}_{\mathcal{S}\backslash k}\big|_{\mathcal{P}^*_{\text{perm},i}}\right) \\
&= \mathbb{E}_{\boldsymbol{d}}\left(\sum_{\substack{\mathcal{S}:\mathcal{S}\subset\mathcal{K}\\ \mathcal{S}\neq\emptyset}} \frac{1}{K!}\sum_{i=1}^{K!}\max_{k\in\mathcal{S}} x^{d_k}_{\mathcal{S}\backslash k}\big|_{\mathcal{P}^*_{\text{perm},i}}\right) \\
&\geq \mathbb{E}_{\boldsymbol{d}}\left(\sum_{\substack{\mathcal{S}:\mathcal{S}\subset\mathcal{K}\\ \mathcal{S}\neq\emptyset}} \max_{k\in\mathcal{S}} \frac{1}{K!}\sum_{i=1}^{K!} x^{d_k}_{\mathcal{S}\backslash k}\big|_{\mathcal{P}^*_{\text{perm},i}}\right) \\
&= \mathbb{E}_{\boldsymbol{d}}\left(\sum_{\substack{\mathcal{S}:\mathcal{S}\subset\mathcal{K}\\ \mathcal{S}\neq\emptyset}} \max_{k\in\mathcal{S}} \bar{x}^{d_k}_{\mathcal{S}\backslash k}\right),
\end{aligned}
$$

where we defined $\bar{x}^n_{\mathcal{S}} = \frac{1}{K!}\sum_{i=1}^{K!} x^n_{\mathcal{S}}\big|_{\mathcal{P}^*_{\text{perm},i}}$ and used the convexity of the $\max$ function. Notice that the RHS is the expected rate when for each file n and subset $\mathcal{S}$, we use the average of the corresponding placement parameters over all permutations of the optimal placement. Because of symmetry, $\bar{x}^n_{\mathcal{S}}$ has the property that $\bar{x}^n_{\mathcal{S}_1} = \bar{x}^n_{\mathcal{S}_2}$ if $|\mathcal{S}_1| = |\mathcal{S}_2|$. From the facts that i) the LHS is the optimal rate, ii) the placement parameters $\bar{x}^n_{\mathcal{S}}$ use the same amount of cache storage as $\mathcal{P}^*$ based on eq. (2) and iii) the sum of $\bar{x}^n_{\mathcal{S}}$ over all subsets $\mathcal{S} \subset [K]$ is 1, we conclude that the rate in the RHS must also be equal to R^*. This implies that there exists an optimal placement with the property that $x^n_{\mathcal{S}} = x^n_s$ for all $\mathcal{S} : |\mathcal{S}| = s$, which completes the proof. ■

APPENDIX B
REVIEW OF SUBMODULAR FUNCTIONS AND ANALYSIS

We review the definition of a submodular set function and present the results that are related to our analysis in Section IV. An extended discussion can be found in [26].

Definition 3: Let $V = \{1, \ldots, p\}$ be a set of p objects. For $\boldsymbol{w} \in \mathbb{R}^p$, $\text{Supp}(\boldsymbol{w}) \subset V$ denotes the support of $\boldsymbol{w}$, defined as $\text{Supp}(\boldsymbol{w}) = \{j \in V, w_j \neq 0\}$.

Definition 4: **(Submodular function)** A set-function $F : 2^V \to \mathbb{R}$ is submodular if and only if, for all subsets $A, B \subset V$, we have $F(A) + F(B) \geq F(A \cup B) + F(A \cap B)$.

Definition 5: **(Lovász extension)** Given a set-function F such that $F(\emptyset) = 0$, the Lovász extension $f : \mathbb{R}^p_+ \to \mathbb{R}$ of F is defined as

$$
f(\boldsymbol{w}) = \sum_{k=1}^{p} w_{j_k}\left[F(\{j_1, \ldots, j_k\}) - F(\{j_1, \ldots, j_{k-1}\})\right],
$$

where $\boldsymbol{w} \in \mathbb{R}^p_+$, $(j_1, \ldots, j_p)$ is a permutation such that $w_{j_1} \geq \ldots \geq w_{j_p}$.

Consider vector $\boldsymbol{\delta} \in \{0,1\}^p$ as the indicator vector for subset $A \subset V$, i.e, for $i \in V$, $\delta_i = 1$ if and only if $i \in A$. Consequently, A is the support of $\boldsymbol{\delta}$. Notice that for the Lovász extension, $f(\boldsymbol{\delta}) = F(\text{Supp}(\boldsymbol{\delta}))$. Hence, f can be seen as an extension of F from vectors in $\{0,1\}^p$ to all vectors in $\mathbb{R}^p_+$. The Lovász extension f has the following properties: 1) it is piecewise-linear, 2) when F is submodular, f is convex, and 3) minimizing F over subsets, i.e., minimizing f over $\{0,1\}^p$, is equivalent to minimizing f over $[0,1]^p$.

Definition 6: **(Stable Sets)** A set A is stable if it cannot be augmented without increasing F, i.e., if for all sets $B \supset A$, $B \neq A$, then $F(B) > F(A)$.

Definition 7: **(Separable Sets)** A set A is separable if we can find a partition of A into $A = B_1 \cup \ldots \cup B_k$ such that $F(A) = F(B_1) + \ldots + F(B_k)$. A set A is inseparable if it is not separable.

Proposition B.1: [26, Section 4.1] For a set of objects V and a nonnegative set-function $d(\cdot)$, let $\Omega(\boldsymbol{w}) = \sum_{G\subset V} d(G)\,\|\boldsymbol{w}_G\|_\infty$ for $\boldsymbol{w} \in \mathbb{R}^p$, where $\boldsymbol{w}_G$ is the subvector of $\boldsymbol{w}$ that only includes entries corresponding to the elements in set G. Function $\Omega(\boldsymbol{w})$ is a norm if $\cup_{G,d(G)>0} G = V$ and it corresponds to the nondecreasing submodular function $F(A) = \sum_{G:A\cap G\neq\emptyset} d(G)$.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIT.2019.2946361, IEEE Transactions on Information Theory

Proposition B.2: [26, Proposition 2]) The extreme points of the unit ball of Ω are the vectors $\frac{1}{F(A)}\boldsymbol{v}$, with $\boldsymbol{v} \in \{-1,0,1\}^p$, $\text{Supp}(\boldsymbol{v}) = A$ and A a stable inseparable set.

APPENDIX C
PROOF OF THEOREM 1

To prove Theorem 1, we first prove the following lemma.

Lemma C.1: The extreme points of the region $[f_{\text{c}} \leq t]^+$ defined by (12b) and (12d) are the origin and points of the form

$$\frac{t}{\frac{K-s}{s+1}\left[1-(1-P(A))^{s+1}\right]}\boldsymbol{v}, \tag{21}$$

where vector $\boldsymbol{v} \in \{0,1\}^{KN}$, $\text{Supp}(\boldsymbol{v}) = A$, and set A is a subset of $[N]_{(s-1)N}$ for an $s \in [K-1]$.

Proof: To obtain the extreme points of $[f_{\text{c}} \leq t]^+$ we begin with the extreme points of the norm-ball $f_{\text{c}} \leq t$ and remove the ones that have negative coordinates as they do not belong to the non-negative orthant. Further, $[f_{\text{c}} \leq t]^+$ has extra extreme points that result from the intersection of $f_{\text{c}} \leq t$ and planes $y_s^n = 0$. Norm-ball f_{c} is symmetric w.r.t. every plane $y_s^n = 0$ and hence the extreme point resulting from the intersection of the norm-ball and such a plane will either be an extreme point of the norm ball or the midpoint of two extreme points of the norm-ball with y_s^n coordinates of $+1$ and -1. In the latter case, the y_s^n coordinate of the extreme point of $[f_{\text{c}} \leq t]^+$ will be 0. Either case, $\text{Supp}(\boldsymbol{v})$ will still be a subset of $[N]_{(s-1)N}$ for an $s \in [K-1]$. If there is no nonzero entry left in the coordinates of the extreme point of the intersection, which is the case when all planes $y_s^n = 0$ intersect, the resulting point is the origin. ■

We now prove Thereom 1.

Proof of Theorem 1: At optimality, we have $f_{\text{c}}(\tilde{\boldsymbol{y}}^*) = t^*$, as otherwise the objective can be decreased by replacing t^* with $f_{\text{c}}(\tilde{\boldsymbol{y}}^*)$, which contradicts the optimality of t^*.

The objective function (12a) calculated at an extreme point of form (21) with nonzero parameters for $s = s_o$ and A is $[1 + \frac{\sum_{n\in g^{-1}(A)}(s_o/K-1+\alpha_n^*)\gamma - Kp_n\alpha_n^*}{(K-s_o)/(s_o+1)(1-(1-P(A))^{s_o+1}}]t$, which is a factor of t. Denote the denominator of (21) by $t^u(s,A)$, i.e., $t^u(s,A) = \frac{K-s}{s+1}[1-(1-P(A))^{s+1}]$. Notice that for $t = t^u(s_o,A)$, the extreme points of $[f_{\text{c}} \leq t]^+$ are of form $y_s^n = 1$ for $s = s_o, n \in g^{-1}(A)$, and $y_s^n = 0$ otherwise. These parameters satisfy (12b)-(12d) and are feasible. Hence, for any $s_o \in [K-1]$ and $A \subset [N]_{(s_o-1)N}$, we have

$$\begin{aligned}
&\left[1+\frac{\sum_{n\in g^{-1}(A)}(\frac{s_o}{K}-1+\alpha_n^*)\gamma - Kp_n\alpha_n^*}{\frac{K-s_o}{s_o+1}(1-(1-P(A))^{s_o+1}}\right]t^u(s_o,A)\\
&\geq t^* + \sum_{n=1}^{N}\sum_{s=1}^{K-1}[(\frac{s}{K}-1+\alpha_n^*)\gamma - Kp_n\alpha_n^*](y_s^n)^*\\
&= \left[1+\frac{\sum_{n\in g^{-1}(A^*)}(\frac{s^*}{K}-1+\alpha_n^*)\gamma - Kp_n\alpha_n^*}{\frac{K-s^*}{s^*+1}(1-(1-P(A^*))^{s^*+1}}\right]t^*
\end{aligned}$$

where the equality holds as the extreme points of $[f_{\text{c}} \leq t^*]^+$ are in the form of (21), and one of them, say $\bar{\boldsymbol{y}}$, with $s = s^*$ and $A = A^*$ has the smallest objective (12a) among the extreme points. Since the inequality holds for every $s_o \in [K-1]$ and $A \subset [N]_{(s_o-1)N}$, it also holds for $s = s^*$ and $A = A^*$ in the LHS. This yields $t^u(s^*,A^*) \geq t^*$ and equivalently $\frac{t^*}{t^u(s^*,A^*)} \leq 1$. As a result, the extreme point $\bar{\boldsymbol{y}}$ also satisfies (12c) and is feasible to (12). Given that it has the smallest objective among the extreme points of $[f_{\text{c}} \leq t^*]^+$, it is optimal, i.e., $\bar{\boldsymbol{y}} = \boldsymbol{y}^*$.

Now, the objective $\left[1+\frac{\sum_{n\in g^{-1}(A^*)}(\frac{s^*}{K}-1+\alpha_n^*)\gamma - Kp_n\alpha_n^*}{\frac{K-s^*}{s^*+1}(1-(1-P(A^*))^{s^*+1}}\right]t^*$ is linear in t^*. Since $t^* \leq t^u(s^*,A^*)$ and $t^* = t^u(s^*,A^*)$ is achievable, at optimality we either have $t^* = 0$ or $t^* = t^u(s^*,A^*)$, depending on the sign of the coefficient of t^*. In the former case, we either have cached all files, i.e., $\forall n : (y_K^n)^* = 1, (y_s^n)^* = 0, s < K$, or no file is cached at all, i.e., $\forall n : (y_0^n)^* = 1, (y_s^n)^* = 0, s > 1$, as in both cases the rate f_{c} due to delivery of the content cached in at least one cache is 0.

In the case of $t^* = t^u(s^*,A^*)$, for $s \in [K-1]$ we have $(y_s^n)^* = 1, s = s^*, n \in g^{-1}(A^*)$ and $(y_s^n)^* = 0$ otherwise. Together with Lemma 1, this concludes that at optimality $(z^n)^* = 1 - \sum_{s=1}^{K-1}(y_s^n)^* \in \{0,1\}$. Hence, when $(z^n)^* = 1$ we have $(y_0^n)^* = 1$ and $(y_K^n)^* = 0$ if $Kp_n < \gamma$ and $(y_0^n)^* = 0$ and $(y_K^n)^* = 1$ if $Kp_n \geq \gamma$. ■

APPENDIX D
PROOF OF LEMMA 3

To prove Lemma 3, we first show the following result:

Lemma D.1: The capacity constraint (7b) in RMSC is satisfied with equality at optimality, i.e., no storage remains unused.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIT.2019.2946361, IEEE Transactions on Information Theory

Proof: Assume that for storage capacity M, there is an optimal solution $\boldsymbol{y}^*$ with $m(\boldsymbol{y}^*)+\epsilon N = M$, where $\epsilon > 0$. Then, construct solution $\boldsymbol{y}'$ with $y'^n_s = (1-\epsilon)y^n_s, s < K$ and $y'^n_K = (1-\epsilon)y^n_K + \epsilon$. Essentially, $\boldsymbol{y}'$ splits every file into two parts of lengths $(1-\epsilon)F$ and ϵF. It uses $\boldsymbol{y}^*$ for the placement of the parts of length $(1-\epsilon)F$ and caches the other ϵF parts on every cache. This uses $(1-\epsilon)m(\boldsymbol{y}^*)+\epsilon \leq (1-\epsilon)M + \epsilon N < M$ of storage, which implies that the storage constraint is satisfied for $\boldsymbol{y}'$. However, $r(\boldsymbol{y}') = (1-\epsilon)r(\boldsymbol{y}^*)+\epsilon\times 0 < r(\boldsymbol{y}^*)$. This contradicts the optimality of $\boldsymbol{y}^*$. Hence, the optimal solution of RMSC must satisfy the capacity constraint by equality. ∎

Proof of Lemma 3: The first property follows from the shadow price interpretation of the Lagrange multipliers for inequality constraints [22, Section 5.6]. In particular, let denote the optimal solutions to (7) with storage budgets M_1 and $M_2 < M_1$ by $\boldsymbol{y}^*_1$ and $\boldsymbol{y}^*_2$. Then, $r(\boldsymbol{y}^*_1) \leq r(\boldsymbol{y}^*_2)$. Since duality gap is zero, the primal and dual objectives are equal, this implies $r(\boldsymbol{y}^*_1)+\gamma^*_1 m(\boldsymbol{y}^*_1) \leq r(\boldsymbol{y}^*_2)+\gamma^*_2 m(\boldsymbol{y}^*_2)$. Hence, $\gamma^*_1 \leq \gamma^*_2$ as otherwise $r(\boldsymbol{y}^*_1)+\gamma^*_2 m(\boldsymbol{y}^*_1) < r(\boldsymbol{y}^*_1)+\gamma^*_1 m(\boldsymbol{y}^*_1) \leq r(\boldsymbol{y}^*_2)+\gamma^*_2 m(\boldsymbol{y}^*_2)$, which contradicts optimality of $\boldsymbol{y}^*_2$.

The second property follows from the fact that the set $\gamma \geq 0$ in the Lagrangian minimization problem (8), or equivalently in JRSM, is continuous, while set $\mathcal{Y}^*$ (or $\mathcal{M}$) is finite. Hence, a range of values of γ must map to the same storage $M \in \mathcal{M}$ and they are all dual optimal.

To prove the third property consider $\boldsymbol{y}_1, \boldsymbol{y}_2 \in \mathcal{Y}^*$ that correspond to two consecutive storage values $M_1 = m(\boldsymbol{y}_1)$ and $M_2 = m(\boldsymbol{y}_2)$. Without loss of generality assume that $M_1 < M_2$. Clearly, $M \notin \mathcal{M}$ for any $M \in (M_1, M_2)$. Now, notice that i) each capacity M must correspond to some γ^* in the dual problem, ii) for each $\gamma \geq 0$ there is an optimal solution to JRSM in $\mathcal{Y}^*$ and a corresponding storage value in $\mathcal{M}$, hence none of those solutions uses an amount of storage $M \notin \mathcal{M}$ and iii) based on Lemma D.1, at optimality all the available storage must be used, and iv) based on property 1, γ^* is nondecreasing in M. These point conclude that the optimal dual parameter for any $M \in [M_1, M_2]$ must belong to $\{\gamma^*_{M_1}, \gamma^*_{M_2}\}$, where γ^*_m represents the optimal dual parameter for capacity m and based on property 1, $\gamma^*_{M_2} \leq \gamma^*_{M_1}$. More specifically, point (iv) requires $\gamma^*_M = \gamma^*_{M_1}$ for $M \in [M_1, M']$ and $\gamma^*_M = \gamma^*_{M_2}$ for $M \in (M', M_2]$, for some $M_1 \leq M' \leq M_2$. However, we must have $\gamma^*_{M_2} = \gamma^*_{M_1}$ as otherwise for any $\gamma^*_{M_2} < \gamma'' < \gamma^*_{M_1}$ corresponds to a value of $M \notin \mathcal{M}$, which contradicts point (ii). This concludes property 3, i.e., for two consecutive values $M_1, M_2 \in \mathcal{M}, M_1 < M_2$, all capacities $M_1 \leq M \leq M_2$ correspond to the same dual parameter γ^*. Further, we can derive the optimal dual parameter for $m(\boldsymbol{y}_1) \leq M \leq \tilde{m}(\boldsymbol{y}_2)$ as it satisfies $r(\boldsymbol{y}_1)+\gamma^* m(\boldsymbol{y}_1) = r(\boldsymbol{y}_2)+\gamma^* m(\boldsymbol{y}_2) = L^*$. Hence,

$$\gamma^* = \frac{r(\boldsymbol{y}_1) - r(\boldsymbol{y}_2)}{m(\boldsymbol{y}_2) - m(\boldsymbol{y}_1)}. \tag{22}$$

∎

APPENDIX E
PROOF OF THEOREM 2

Proof: To prove Theorem 2, we consider two cases:

Case I ($M \in \mathcal{M}$): This case is straightforward because of the zero duality gap in the primal-dual framework established in Section III-B. In particular, vector $\boldsymbol{y}^*_{\text{JRSM}} \in \mathcal{Y}^*$ with $m(\boldsymbol{y}^*_{\text{JRSM}}) = M$ is also optimal to RMSC. [10]

Case II ($M \notin \mathcal{M}$): To derive the optimal solution of RMSC for $M \notin \mathcal{M}$, we use Lemma 3. Let $\boldsymbol{y}_1, \boldsymbol{y}_2 \in \mathcal{Y}^*$ be the solutions corresponding to the two consecutive storage values $m(\boldsymbol{y}_1), m(\boldsymbol{y}_2) \in \mathcal{M}$ such that $m(\boldsymbol{y}_1) < M < m(\boldsymbol{y}_2)$. Let γ^* and L^* be the corresponding optimal dual parameter and Lagrangian value, respectively. Since $m(\cdot)$ is linear, for any given storage $m(\boldsymbol{y}_1) < M < m(\boldsymbol{y}_2)$, there exists a convex combination of $\boldsymbol{y}_1$ and $\boldsymbol{y}_2$ that uses storage M. We show that the same convex combination also minimizes the Lagrangian for γ^*. In that case, we are back to a case similar to Case I, and the same argument used there requires the convex combination to also optimize RMSC.

Consider $\boldsymbol{y}_{\theta;\boldsymbol{y}_1,\boldsymbol{y}_2} \triangleq \theta\boldsymbol{y}_1 + (1-\theta)\boldsymbol{y}_2$ for $0 < \theta < 1$. For the θ for which $m(\boldsymbol{y}_{\theta;\boldsymbol{y}_1,\boldsymbol{y}_2}) = M$, we need to show that $\boldsymbol{y}_{\theta;\boldsymbol{y}_1,\boldsymbol{y}_2}$ minimizes the Lagrangian for γ^*. This is equivalent to showing

[10] A direct proof for optimality of $\boldsymbol{y}^*_{\text{JRSM}}$ with $m(\boldsymbol{y}^*_{\text{JRSM}}) = M$ for RMSC is as follows. Based on Lemma D.1, the optimal solution of RMSC satisfies the capacity constraint with equality. Now, assume that $\boldsymbol{y}^*_{\text{JRSM}}$ is not optimal for the RMSC problem. This means that $r(\boldsymbol{y}^*_{\text{RMSC}}) < r(\boldsymbol{y}^*_{\text{JRSM}})$. However, since $m(\boldsymbol{y}^*_{\text{RMSC}}) = m(\boldsymbol{y}^*_{\text{JRSM}}) = M$, this implies that $r(\boldsymbol{y}^*_{\text{RMSC}}) + \gamma^* m(\boldsymbol{y}^*_{\text{RMSC}}) < r(\boldsymbol{y}^*_{\text{JRSM}}) + \gamma^* m(\boldsymbol{y}^*_{\text{JRSM}})$. The last result contradicts the optimality of $\boldsymbol{y}^*_{\text{JRSM}}$ for JRSM. Hence, $\boldsymbol{y}^*_{\text{JRSM}}$ must be optimal for the RMSC problem.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIT.2019.2946361, IEEE Transactions on Information Theory

that $r(\boldsymbol{y}_{\theta;\boldsymbol{y}_1,\boldsymbol{y}_2}) + \gamma^* m(\boldsymbol{y}_{\theta;\boldsymbol{y}_1,\boldsymbol{y}_2}) = L^*$.[11] Since $m(\boldsymbol{y}_{\theta;\boldsymbol{y}_1,\boldsymbol{y}_2}) = \theta m(\boldsymbol{y}_1)+(1-\theta)m(\boldsymbol{y}_2)$, it is sufficient to that $r(\boldsymbol{y}_{\theta;\boldsymbol{y}_1,\boldsymbol{y}_2}) = \theta r(\boldsymbol{y}_1) + (1-\theta)r(\boldsymbol{y}_2)$ to prove $r(\boldsymbol{y}_{\theta;\boldsymbol{y}_1,\boldsymbol{y}_2}) + \gamma^* m(\boldsymbol{y}_{\theta;\boldsymbol{y}_1,\boldsymbol{y}_2}) = L^*$. To show $r(\boldsymbol{y}_{\theta;\boldsymbol{y}_1,\boldsymbol{y}_2}) = \theta r(\boldsymbol{y}_1)+(1-\theta)r(\boldsymbol{y}_2)$, notice that each $\boldsymbol{y} \in \mathcal{Y}^*$ has nonzero parameters y_s^n for at most two values of s, one $s = 0$ and one $s \geq 1$. Assume that $\boldsymbol{y}_1$ and $\boldsymbol{y}_2$ have nonzero entries respectively for $s_1 > 0$ and $s_2 > 0$ and possibly for $s = 0$. Let n_1 and n_2 be the largest indexes n with nonzero $y_{s_1}^n$ and $y_{s_2}^n$ in the respective two solutions. Notice that since $\boldsymbol{y}_i \in \mathcal{Y}^*$, having $y_{s_i}^n > 0$ implies $y_{s_i}^n = 1$. We consider two cases of $s_1 \neq s_2$ and $s_1 = s_2$. In the former case, $r(\boldsymbol{y}_1) = K\sum_{n=1}^N p_n y_{1\,0}^n + \frac{K-s_1}{s_1+1}(1-(1-\sum_{n=1}^{n_1} p_n)^{s_1+1})$, $r(\boldsymbol{y}_2) = K\sum_{n=1}^N p_n y_{2\,0}^n + \frac{K-s_2}{s_2+1}(1-(1-\sum_{n=1}^{n_2} p_n)^{s_2+1})$ and $r(\theta\boldsymbol{y}_1 + (1-\theta)\boldsymbol{y}_2)) = K\sum_{n=1}^N p_n(\theta y_{1\,0}^n+(1-\theta)y_{2\,0}^n)+\frac{K-s_1}{s_1+1}(1-(\sum_{n=1}^{n_1} p_n)^{s_1+1})\theta + \frac{K-s_2}{s_2+1}(1-(\sum_{n=1}^{n_2} p_n)^{s_2+1})(1-\theta) = \theta\tilde{r}(\boldsymbol{y}_1) + (1-\theta)r(\boldsymbol{y}_2)$. For the case of $s_1 = s_2 = s_o$, since $m(\boldsymbol{y}_1) < m(\boldsymbol{y}_2)$, we must have $n_2 > n_1$. Hence, for the rates we have

$$\begin{aligned}
r(\boldsymbol{y}_1) &= K\sum_{n=1}^N p_n y_{0\,1}^n + \frac{K-s_o}{s_o+1}\sum_{g\in\mathcal{G}_{s_o+1}} \pi_{s_o+1}^g \max_{n\in g} y_{s_o 1}^n \\
&= K\sum_{n=1}^N p_n y_{0\,1}^n + \frac{K-s_o}{s_o+1}\sum_{g\in\mathcal{G}_{s_o+1}, g\cap[n_1]\neq\emptyset} \pi_{s_o+1}^g \\
r(\boldsymbol{y}_2) &= K\sum_{n=1}^N p_n y_{0\,2}^n + \frac{K-s_o}{s_o+1}\sum_{g\in\mathcal{G}_{s_o+1}} \pi_{s_o+1}^g \max_{n\in g} y_{s_o 2}^n \\
&= K\sum_{n=1}^N p_n y_{0\,2}^n + \frac{K-s_o}{s_o+1}\sum_{g\in\mathcal{G}_{s_o+1}, g\cap[n_2]\neq\emptyset} \pi_{s_o+1}^g
\end{aligned}$$

and

$$\begin{aligned}
&r(\theta\boldsymbol{y}_1 + (1-\theta)\boldsymbol{y}_2) \\
&= K\sum_{n=1}^N p_n(\theta y_{0\,1}^n + (1-\theta)y_{0\,2}^n) \\
&\quad + \frac{K-s_o}{s_o+1}\sum_{g\in\mathcal{G}_{s_o+1}} \pi_{s_o+1}^g \max_{n\in g} \theta y_{s_o 1}^n + (1-\theta)y_{s_o 2}^n \\
&= K\sum_{n=1}^N p_n(\theta y_{0\,1}^n + (1-\theta)y_{0\,2}^n) \\
&\quad + \frac{K-s_o}{s_o+1}\sum_{g\in\mathcal{G}_{s_o+1}, g\cap[n_1]\neq\emptyset} \pi_{s_o+1}^g(\theta + (1-\theta)) \\
&\quad + \frac{K-s_o}{s_o+1}\sum_{g\in\mathcal{G}_{s_o+1}, g\cap[n_1]=\emptyset, g\cap[n_2-n_1]_{n_1}\neq\emptyset} \pi_{s_o+1}^g(1-\theta) \\
&= \theta\left[K\sum_{n=1}^N p_n y_{1\,0}^n + \frac{K-s_o}{s_o+1}\sum_{g\in\mathcal{G}_{s_o+1}, g\cap[n_1]\neq\emptyset} \pi_{s_o+1}^g\right] \\
&\quad + (1-\theta)\left[K\sum_{n=1}^N p_n y_{2\,0}^n + \frac{K-s_o}{s_o+1}\sum_{g\in\mathcal{G}_{s_o+1}\emptyset, g\cap[n_2]\neq\emptyset} \pi_{s_o+1}^g\right] \\
&= \theta\tilde{r}(\boldsymbol{y}_1) + (1-\theta)r(\boldsymbol{y}_2)
\end{aligned}$$

where we used the fact that if $g \cap [n_1] \neq \emptyset$, then $n_2 > n_1$ implies $g \cap [n_2] \neq \emptyset$. This completes the proof of the third feature as we now have

$$\begin{aligned}
&r(\boldsymbol{y}_{\theta;\boldsymbol{y}_1,\boldsymbol{y}_2}) + \gamma^* m(\boldsymbol{y}_{\theta;\boldsymbol{y}_1,\boldsymbol{y}_2}) \\
&= \theta\tilde{r}(\boldsymbol{y}_1) + (1-\theta)r(\boldsymbol{y}_2) + \gamma^*[\theta\tilde{m}(\boldsymbol{y}_1) + (1-\theta)m(\boldsymbol{y}_2)] \\
&= \theta[r(\boldsymbol{y}_1) + \gamma^* m(\boldsymbol{y}_1)] + (1-\theta)[r(\boldsymbol{y}_2) + \gamma^* m(\boldsymbol{y}_2)] \\
&= \theta L^* + (1-\theta)L^* = L^*.
\end{aligned}$$

■

[11] The equivalence results from the fact that if $r(\boldsymbol{y}_{\theta;\boldsymbol{y}_1,\boldsymbol{y}_2}) + \gamma^* m(\boldsymbol{y}_{\theta;\boldsymbol{y}_1,\boldsymbol{y}_2}) < L^*$, then $\boldsymbol{y}_1$ and $\boldsymbol{y}_2$ could not be optimal for γ^*, which is a contradiction.

REFERENCES

[1] D. Liu, B. Chen, C. Yang, and A. F. Molisch, "Caching at the wireless edge: design aspects, challenges, and future directions," *IEEE Commun. Mag.*, vol. 54, no. 9, pp. 22–28, Sep. 2016.

[2] E. Zeydan, E. Bastug, M. Bennis, M. A. Kader, I. A. Karatepe, A. S. Er, and M. Debbah, "Big data caching for networking: moving from cloud to edge," *IEEE Commun. Mag.*, vol. 54, no. 9, pp. 36–42, Sep. 2016.

[3] N. Zhao, X. Liu, F. R. Yu, M. Li, and V. C. M. Leung, "Communications, caching, and computing oriented small cell networks with interference alignment," *IEEE Commun. Mag.*, vol. 54, no. 9, pp. 29–35, Sep. 2016.

[4] K. Shanmugam, N. Golrezaei, A. G. Dimakis, A. F. Molisch, and G. Caire, "Femtocaching: Wireless content delivery through distributed caching helpers," *IEEE Trans. Inf. Theory*, vol. 59, no. 12, pp. 8402–8413, Dec. 2013.

[5] M. K. Kiskani and H. R. Sadjadpour, "Multihop caching-aided coded multicasting for the next generation of cellular networks," *IEEE Transactions on Vehicular Technology*, vol. 66, no. 3, pp. 2576–2585, March 2017.

[6] N. Pappas, Z. Chen, and I. Dimitriou, "Throughput and delay analysis of wireless caching helper systems with random availability," *IEEE Access*, vol. 6, pp. 9667–9678, 2018.

[7] J. Song, H. Song, and W. Choi, "Optimal content placement for wireless femto-caching network," *IEEE Transactions on Wireless Communications*, vol. 16, no. 7, pp. 4433–4444, July 2017.

[8] M. A. Maddah-Ali and U. Niesen, "Coding for caching: fundamental limits and practical challenges," *IEEE Commun. Mag.*, vol. 54, no. 8, pp. 23–29, Aug. 2016.

[9] ——, "Fundamental limits of caching," *IEEE Trans. Inf. Theory*, vol. 60, no. 5, pp. 2856–2867, May 2014.

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

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

[12] J. Hachem, N. Karamchandani, and S. Diggavi, "Content caching and delivery over heterogeneous wireless networks," in *IEEE Conference on Computer Communications (INFOCOM)*, Apr. 2015, pp. 756–764.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIT.2019.2946361, IEEE Transactions on Information Theory

[13] M. Ji, A. M. Tulino, J. Llorca, and G. Caire, "On the average performance of caching and coded multicasting with random demands," in *Proc. 11th International Symposium on Wireless Communications Systems (ISWCS)*, Aug. 2014, pp. 922–926.

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

[15] S. Jin, Y. Cui, H. Liu, and G. Caire, "Structural properties of uncoded placement optimization for coded delivery," *CoRR*, vol. abs/1707.07146, 2017. [Online]. Available: http://arxiv.org/abs/1707.07146

[16] A. M. Daniel and W. Yu, "Optimization of heterogeneous coded caching," *CoRR*, vol. abs/1708.04322, 2017. [Online]. Available: http://arxiv.org/abs/1708.04322

[17] S. A. Saberali, H. E. Saffar, L. Lampe, and I. F. Blake, "Adaptive delivery in caching networks," *CoRR*, vol. abs/1707.09662, 2017. [Online]. Available: http://arxiv.org/abs/1707.09662

[18] S. Sahraei, P. Quinton, and M. Gastpar, "The optimal memory-rate trade-off for the non-uniform centralized caching problem with two files under uncoded placement," *IEEE Trans. Inf. Theory*, pp. 1–1, 2019.

[19] 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.

[20] Z. Bar-Yossef, Y. Birk, T. S. Jayram, and T. Kol, "Index coding with side information," *IEEE Trans. Inf. Theory*, vol. 57, no. 3, pp. 1479–1494, Mar. 2011.

[21] J. Hachem, N. Karamchandani, and S. N. Diggavi, "Effect of number of users in multi-level coded caching," in *Proc. IEEE Int. Symp. Information Theory*, June 2015, pp. 1701–1705.

[22] S. Boyd and L. Vandenberghe, *Convex Optimization*. New York, NY, USA: Cambridge University Press, 2004.

[23] D. G. Luenberger and Y. Ye, *Linear and Nonlinear Programming*. Springer Publishing Company, Incorporated, 2015.

[24] S. A. Saberali, H. Ebrahimzadeh Saffar, L. Lampe, and I. Blake, "Adaptive delivery in caching networks," *IEEE Communications Letters*, vol. 20, no. 7, pp. 1405–1408, July 2016.

[25] Q. Yu, M. A. Maddah-Ali, and A. S. Avestimehr, "The exact rate-memory tradeoff for caching with uncoded prefetching," *IEEE Trans. Inf. Theory*, vol. 64, no. 2, pp. 1281–1296, Feb 2018.

[26] F. R. Bach, "Structured sparsity-inducing norms through submodular functions," in *Advances in Neural Information Processing Systems*, 2010, pp. 118–126.

Seyed Ali Saberali received his B.Sc. degree from Isfahan University of Technology, Iran, in 2011, and his M.Sc. degree from the University of Alberta, Canada, in 2013, both in electrical and computer engineering. He is pursuing the Ph.D. degree in electrical and computer engineering at UBC. His research interests include coding theory, machine learning and optimization theory.

Lutz Lampe (M'02-SM'08) received the Dipl.-Ing. and Dr.-Ing. degrees in electrical engineering from the University of Erlangen, Germany, in 1998 and 2002, respectively.

Since 2003, he has been with the Department of Electrical and Computer Engineering, The University of British Columbia, Vancouver, BC, Canada, where he is a Full Professor. His research interests are broadly in theory and application of wireless, power line, optical wireless and optical fibre communications.

Dr. Lampe is currently an Associate Editor for the IEEE TRANSACTIONS ON COMMUNICATIONS, the IEEE COMMUNICATIONS LETTERS, and the IEEE COMMUNICATIONS SURVEYS AND TUTORIALS. He was a (co-)recipient of a number of best paper awards, including awards at the 2006 IEEE International Conference on Ultra-Wideband (ICUWB), the 2010 IEEE International Communications Conference (ICC), and the 2011, 2017 and 2018 IEEE International Conference on Power Line Communications and Its Applications (ISPLC). He was the General (Co-)Chair for the 2005 IEEE ISPLC, the 2009 IEEE ICUWB and the 2013 IEEE International Conference on Smart Grid Communications (SmartGridComm). He is a co-editor of the book "Power Line Communications: Principles, Standards and Applications from Multimedia to Smart Grid," published by John Wiley & Sons in its 2nd edition in 2016.

Ian F. Blake (M'65–F'–LF'06) received his undergraduate education at Queen's University in Kingston, Ontario and his Ph.D. at Princeton University in New Jersey. From 1967 to 1969 he was a Research Associate with the Jet Propulsion Laboratories in Pasadena, California. From 1969 to 1996 he was with the Department of Electrical and Computer Engineering at the University of Waterloo, in Waterloo, Ontario where he was Chairman from 1978 to 1984 and Director of the Institute of Computer Research from 1990 to 1994. He is currently an Honorary Professor at the Department of Electrical and Computer Engineering at the University of British Columbia.

He has spent sabbatical leaves with the IBM Thomas J. Watson Research Center, the IBM Research Laboratories in Switzerland and M/A-Com Linkabit in San Diego, California. From 1996-1999 he was with the Hewlett-Packard Labs in Palo Alto, California. His research interests are in the areas of cryptography, algebraic coding theory, digital communications and spread spectrum systems. He is a Fellow of the IEEE, the Royal Society of Canada and the Canadian Academy of Engineers. He was awarded an IEEE Millenium Medal and the Aaron D. Wyner Distinguished Service Award of the IEEE Information Theory Society.