A General Coded Caching Scheme for Scalar Linear Function Retrieval

Yinbin Ma and Daniela Tuninetti,
University of Illinois Chicago, Chicago, IL 60607, USA
Email:{yma52, danielat}@uic.edu

***Abstract*—Coded caching aims to minimize the network's peak-time communication load by leveraging the information pre-stored in the local caches at the users. The original single file retrieval setting by Maddah-Ali and Niesen has been recently extended to general Scalar Linear Function Retrieval (SLFR) by Wan *et al.*, who proposed a linear scheme that surprisingly achieves the same optimal load (under the constraint of uncoded cache placement) as in single file retrieval. This paper's goal is to characterize the conditions under which a general SLFR linear scheme is optimal and gain practical insights into why the specific choices made by Wan *et al.* work. This paper shows that the optimal decoding coefficients are necessarily the product of two terms, one only involving the encoding coefficients and the other only the demands. In addition, the relationships among the encoding coefficients are shown to be captured by the cycles of a certain graph. Thus, a general linear scheme for SLFR can be found by solving a spanning tree problem.**

A full version of this paper can be found at [1].

I. Introduction

Coded caching, originally introduced by Maddah-Ali and Niesen (MAN) in [2], has been the focus of much research efforts recently as it predicts, for networks with a server delivering a single file to each cache-aided user, that it is possible to achieve a communication load that does not scale with the number of users. Yu *et al.* in [3] improved on the delivery phase of the MAN scheme by removing the MAN multicast message transmissions that are redundant when a file is requested by multiple users, and thus showed that the converse bound under the constraint of uncoded cache placement by Wan *et al.* in [4] is tight. Wan *et al.* in [5] recently extended the MAN setup so as to allow users to request general scalar linear combinations of the files stored at the server. Despite the fact that the number of possible demands increases exponentially in the number of files, [5] surprisingly showed that the optimal communication load is the same as for the single file retrieval setting, at least under uncoded cache placement.

The scheme proposed in [5] is linear. As in [3], the server selects of set of leaders (whose demand vectors are a linearly independent spanning set of the set of all possible demands) and creates multicast messages by performing linear combinations of demanded subfiles that were not cached; the coefficients for such linear combinations are referred to as *encoding coefficients* and can be optimized. As in [3], multicast messages that would only be useful for non-leader users are not sent and have to be locally reconstructed as linear combinations of sent multicast messages; the coefficients for such linear combinations are referred to as *decoding coefficients* and must guarantee that each user correctly decodes its demanded linear combination of files. The choice of encoding and decoding coefficients in [5] is rather non trivial and not a simple extension of [3], which actually fails to guarantee successful decoding on finite fields of characteristics strictly larger than two. The encoding coefficients chosen in [5], inspired by private function retrieval in [6], all have unit modulo but alternate in sign among leaders and among non-leaders. Such a choice works (with corresponding decoding coefficients given, up to a sign, by determinants of certain matrices derived from the demand matrix) but the reason why it is so could not be explained.

This paper aims to gain insights into why the choices in [5] work by analyzing the most general linear scheme (i.e., general encoding and decoding coefficients). Our main contribution is to show that the optimal decoding coefficients are necessarily the product of two terms, one only involving the encoding coefficients and the other only the determinants of certain matrices derived from the demands. In addition, we characterize the relationships the encoding coefficients need to satisfy in order to guaranteed successful decoding as cycles on a certain graph. *Thus, we show that a general SLFR linear scheme can be found by solving a spanning tree problem.*

The rest of the paper is organized as follow. Section II introduces the cache-aided scalar linear function retrieval (SLFR) problem and summarizes related work. Section III presents our main result, which is proved in Section IV. Section V concludes the paper. Some examples and missing proof details (for sake of space) can be found in [1].

In this paper we use the following notation convention.

- Calligraphic symbols denote sets, bold symbols vectors, and sans-serif symbols system parameters.
- $|\cdot|$ is the cardinality of a set or the length of a vector.
- $\det(M)$ is the determinant of the matrix M.
- $1_{\{\mathcal{E}\}}$ is the indicator function of the event $\mathcal{E}$.
- $M[\mathcal{Q},\mathcal{S}]$ is the submatrix of M obtained by selecting the rows indexed by $\mathcal{Q}$ and the columns indexed by $\mathcal{S}$.
- For an integer b, we let $[b] := \{1,\ldots,b\}$.
- For a ground set $\mathcal{G}$ and an integer t, we let $\Omega^t_{\mathcal{G}} := \{\mathcal{T} \subseteq \mathcal{G} : |\mathcal{T}| = t\}$. Moreover, $\mathcal{S} \setminus \mathcal{Q} := \{k : k \in \mathcal{S}, k \notin \mathcal{Q}\}$.
- $\mathsf{Ind}_{\mathcal{S},k}$ returns the position of the element $k \in \mathcal{S}$, where the element of the integer set $\mathcal{S}$ are considered in increasing order. For example, $\mathsf{Ind}_{\{3,5\},3} = 1$ and $\mathsf{Ind}_{\{3,5\},5} = 2$. By convention $\mathsf{Ind}_{\mathcal{S},k} = 0$ if $k \notin \mathcal{S}$.

II. Problem Formulation and Known Results

A. Problem Formulation

A $(\mathsf{K},\mathsf{N},\mathsf{q},\mathsf{M},\mathsf{R})$ SLFR problem has one central server that has access to a library of N files (denoted as $F_1,\ldots,F_{\mathsf{N}}$), each of B independent and uniformly distributed symbols over the finite field $\mathbb{F}_{\mathsf{q}}$, for some prime-power q. The server communicates through an error-free shared link at *load* R to K users, where each has a local *memory* to store up to M files. The worst-case load $\mathsf{R}^\star(\mathsf{M}), \mathsf{M}\in[0,\mathsf{N}]$, for the SLFR problem is defined as in [5], which is not explicitly written here for sake of space (as it also appears next).

B. Known Results

It was shown in [5] that requesting arbitrary scalar linear functions of the files from the server does not incur any load penalty compared to the case of requesting a single file, that is, the lower convex envelope of the following points is achievable

$$(\mathsf{M},\mathsf{R})=\left(\frac{\mathsf{N}t}{\mathsf{K}},\frac{\binom{\mathsf{K}}{t+1}-\binom{\mathsf{K}-\min(\mathsf{N},\mathsf{K})}{t+1}}{\binom{\mathsf{K}}{t}}\right),\forall t\in[0:\mathsf{K}]. \quad (1)$$

Moreover, the tradeoff in (1) is optimal among all schemes with uncoded cache placement [3], [5] and to within a factor two otherwise [7]. The scheme in [5] is as follows.

a) Cache Placement: Partition the position indices as

$$[\mathsf{B}]=\left\{\mathcal{I}_{\mathcal{T}}:\mathcal{I}_{\mathcal{T}}\subseteq[\mathsf{B}],\mathcal{T}\in\Omega^t_{[\mathsf{K}]},\ |\mathcal{I}_{\mathcal{T}}|=\mathsf{B}/\binom{\mathsf{K}}{t}\right\}, \quad (2)$$

and define (with a Matlab-like notation) the sub-files as

$$F_{i,\mathcal{T}}:=F_i(\mathcal{I}_{\mathcal{T}})\in\mathbb{F}_{\mathsf{q}}^{\mathsf{B}/\binom{\mathsf{K}}{t}},\ \forall\mathcal{T}\in\Omega^t_{[\mathsf{K}]},\ \forall i\in[\mathsf{N}]. \quad (3)$$

The cache of user $k\in[\mathsf{K}]$ is populated as

$$Z_k=\{F_{i,\mathcal{T}}:\mathcal{T}\in\Omega^t_{[\mathsf{K}]},k\in\mathcal{T},i\in[\mathsf{N}]\}\in\mathbb{F}_{\mathsf{q}}^{\mathsf{BN}\binom{\mathsf{K}-1}{t-1}/\binom{\mathsf{K}}{t}}. \quad (4)$$

The memory size is thus $\mathsf{M}=\mathsf{N}\binom{\mathsf{K}-1}{t-1}/\binom{\mathsf{K}}{t}=\mathsf{N}\,t/\mathsf{K}$ as in (1).

b) Delivery: The demand of user $k\in[\mathsf{K}]$ is represented by the row vector $\mathbf{d}_k=(d_{k,1},\ldots,d_{k,\mathsf{N}})\in\mathbb{F}_{\mathsf{q}}^{\mathsf{N}}$, meaning that he needs to successfully retrieve the scalar linear function (i.e., operations are element-wise across files)

$$B_k:=d_{k,1}F_1+\ldots+d_{k,\mathsf{N}}F_{\mathsf{N}}\in\mathbb{F}_{\mathsf{q}}^{\mathsf{B}},\ \forall k\in[\mathsf{K}]. \quad (5)$$

As for the sub-files in (3), we define the *demand-blocks* as

$$B_{k,\mathcal{T}}=B_k(\mathcal{I}_{\mathcal{T}})\in\mathbb{F}_{\mathsf{q}}^{\mathsf{B}/\binom{\mathsf{K}}{t}},\ \forall\mathcal{T}\in\Omega^t_{[\mathsf{K}]},\ \forall k\in[\mathsf{K}]. \quad (6)$$

Some demand-blocks can be computed based on the cache content available locally at the users in (4), while the remaining ones need to be delivered by the server. Let $\mathbb{D}:=[\mathbf{d}_1;\ldots;\mathbf{d}_{\mathsf{K}}]\in\mathbb{F}_{\mathsf{q}}^{\mathsf{K}\times\mathsf{N}}$ be the *demand matrix*. Let $\mathcal{L}\subseteq[\mathsf{K}]$ such that $\mathrm{rank}_{\mathsf{q}}(\mathbb{D})=\mathrm{rank}_{\mathsf{q}}(\mathbb{D}[\mathcal{L},:])=|\mathcal{L}|=:r$ be the *leader set*, which is not unique but its size is (as every finite-dimensional vector space has a basis). Let $\mathbb{D}'\in\mathbb{F}_{\mathsf{q}}^{\mathsf{K}\times|\mathcal{L}|}$ denote the *transformed demand matrix* defined as

$$[\mathbb{D}']_{k,\ell}=\begin{cases}1_{\{k=\ell\}} & \text{if } k\in\mathcal{L}\\ x_{k,\ell} & \text{if } k\notin\mathcal{L}\end{cases},\ \forall k\in[\mathsf{K}],\ \forall\ell\in\mathcal{L}, \quad (7)$$

i.e., the demand-blocks of non-leaders in (5) are expressed as a linear combination of the demand-blocks of the leaders as

$$B_{k,\mathcal{T}}=\sum_{\ell\in\mathcal{L}}x_{k,\ell}\ B_{\ell,\mathcal{T}},\forall\mathcal{T}\in\Omega^t_{[\mathsf{K}]},\ \forall k\in[\mathsf{K}]\backslash\mathcal{L}, \quad (8)$$

where the existence of the coefficients $\{x_{u,\ell}\in\mathbb{F}_{\mathsf{q}}:u\in\overline{\mathcal{L}},\ell\in\mathcal{L}\}$ in (8) follows from linear algebra. The server forms the following multicast messages

$$W_{\mathcal{S}}=\sum_{k\in\mathcal{S}}\alpha_{k,\mathcal{S}\backslash\{k\}}\ B_{k,\mathcal{S}\backslash\{k\}}\in\mathbb{F}_{\mathsf{q}}^{\mathsf{B}/\binom{\mathsf{K}}{t}},\ \forall\mathcal{S}\in\Omega^{t+1}_{[\mathsf{K}]}, \quad (9)$$

for some *encoding coefficients*

$$\{\alpha_{k,\mathcal{S}\backslash\{k\}}\in\mathbb{F}_{\mathsf{q}}\setminus\{0\}:\ k\in[\mathsf{K}],\mathcal{S}\in\Omega^{t+1}_{[\mathsf{K}]}\}. \quad (10)$$

The server sends all multicast messages in (9) that are useful for the leaders, that is, it sends $X\in\mathbb{F}_{\mathsf{q}}^{\Delta+\mathsf{B}(\binom{\mathsf{K}}{t+1}-\binom{\mathsf{K}-|\mathcal{L}|}{t+1})/\binom{\mathsf{K}}{t}}$

$$X=\{W_{\mathcal{S}}:\mathcal{S}\in\Omega^{t+1}_{[\mathsf{K}]},|\mathcal{S}\cap\mathcal{L}|>0\}\cup\{\mathcal{L},\mathbb{D}'\}. \quad (11)$$

Note that sending the chosen leader set and the transformed demand matrix requires $\Delta=|\mathcal{L}|\lceil\log_{\mathsf{q}}(\mathsf{K})\rceil+\mathsf{K}+|\mathcal{L}|$ symbols, where Δ does not scale with the file length B. The worst-case load is thus for $r=|\mathcal{L}|=\min(\mathsf{K},\mathsf{N})$ and equals R in (1).

For a given $\mathcal{S}\in\Omega^{t+1}_{[\mathsf{K}]}$, user $k\in\mathcal{S}$ can decode the missing demand-block $B_{k,\mathcal{S}\backslash\{k\}}$ from $W_{\mathcal{S}}$. The multicast messages $\{W_{\mathcal{A}}:\mathcal{A}\in\Omega^{t+1}_{[\mathsf{K}]\backslash\mathcal{L}}\}$ must be locally reconstructed from the transmitted ones in (11) so that each user can recover all its missing demand-blocks. For $\mathsf{K}-r\geq t+1$, we seek to express

$$W_{\mathcal{A}}=\sum_{\mathcal{S}\in\Omega^{t+1}_{[\mathsf{K}]},|\mathcal{S}\cap\mathcal{L}|>0}\beta_{\mathcal{S}}^{(\mathcal{A})}\ W_{\mathcal{S}},\ \forall\mathcal{A}\in\Omega^{t+1}_{[\mathsf{K}]\backslash\mathcal{L}}, \quad (12)$$

by an appropriate choice of the *decoding coefficients*

$$\{\beta_{\mathcal{S}}^{(\mathcal{A})}\in\mathbb{F}_{\mathsf{q}}:\mathcal{S}\in\Omega^{t+1}_{[\mathsf{K}]},|\mathcal{S}\cap\mathcal{L}|>0,\ \mathcal{A}\in\Omega^{t+1}_{[\mathsf{K}]\backslash\mathcal{L}}\}. \quad (13)$$

The choice of decoding coefficients must work for all realizations of the demand-blocks[1].

In [5] it was proposed to alternate between ±1 the encoding coefficients in (9) as

$$\alpha_{k,\mathcal{S}\backslash\{k\}}=(-1)^{\mathsf{Ind}_{\mathcal{S}\cap\mathcal{L},k}+\mathsf{Ind}_{\mathcal{S}\backslash\mathcal{L},k}},\forall k\in\mathcal{S}, \quad (14)$$

which results in decoding coefficients that are equal, up to a sign, to determinants of certain sub-matrices of $\mathbb{D}'$ in (7). A reason for the choice of alternating signs in (14) (and the resulting decoding coefficients) was not given in [5]. The open question is whether such a choice is fundamental.

We answer this open question by analyzing a general linear scheme in the form of (9) and (12). We show that: (1) the signs of the encoding coefficients must follow a pattern where they alternate, but not necessarily as in (14), and their modulo need not be one; (2) the decoding coefficients are proportional to the determinants of certain matrices obtained from the transformed demand matrix, but the proportionality coefficient need not have modulo one; and, finally and importantly, (3) the encoding and decoding coefficients must satisfy certain relationships that are captured by the cycles of a graph.

[1]The leader set $\mathcal{L}$, the encoding coefficients in (10) and the decoding coefficients in (13) are a function of $\mathbb{D}$ in general; such a dependency is not made explicit here in order not to clutter the notation.

III. Main Result

Our main result is to show that the linear scheme in (9) and (12) is correct if and only if the following holds.

The local reconstruction of non-sent multicast messages in (12) simplifies to solving

$$0=\sum_{\mathcal{S}\in\Omega^{t+1}_{\mathcal{A}\cup\mathcal{L}}}\beta^{(\mathcal{A})}_{\mathcal{S}}W_{\mathcal{S}}:\ \beta^{(\mathcal{A})}_{\mathcal{A}}=-1,\ \forall\mathcal{A}\in\Omega^{t+1}_{[\mathsf{K}]\setminus\mathcal{L}},\quad(15)$$

where in (15) the summation is over subsets of $\mathcal{A}\cup\mathcal{L}$ (in total $\binom{|\mathcal{L}|+t+1}{t+1}$ terms in (15)) rather than over some subsets of $[\mathsf{K}]$ (in total $\binom{\mathsf{K}}{t+1}-\binom{\mathsf{K}-|\mathcal{L}|}{t+1}$ terms in (12)). Eq(15) is solved, for any realization of the files, by using decoding coefficients

$$\beta^{(\mathcal{A})}_{\mathcal{S}}=\widetilde{\beta}^{(\mathcal{A})}_{\mathcal{S}}\cdot\det\left(\mathbb{D}'[\mathcal{A}\setminus\mathcal{S},\mathcal{S}\setminus\mathcal{A}]\right),\quad(16a)$$

$$\forall\mathcal{S}\in\Omega^{t+1}_{\mathcal{A}\cup\mathcal{L}},\ \forall\mathcal{A}\in\Omega^{t+1}_{[\mathsf{K}]\setminus\mathcal{L}},\quad(16b)$$

where the part of the decoding coefficients that does not depend on the demands (denoted as $\widetilde{\beta}^{(\mathcal{A})}_{\{k\}\cup\mathcal{T}}$ next) and the encoding coefficients (denoted as $\alpha_{k,\mathcal{T}}$ next) must satisfy

$$\widetilde{\beta}^{(\mathcal{A})}_{\{k\}\cup\mathcal{T}}\cdot\alpha_{k,\mathcal{T}}=(-1)^{\phi^{(\mathcal{A})}_{k,\mathcal{T}}}\cdot\mathsf{c}^{(\mathcal{A})}_{\mathcal{T}},\quad(17a)$$

$$\phi^{(\mathcal{A})}_{k,\mathcal{T}}=\begin{cases}1+\mathsf{Ind}_{(\{k\}\cup\mathcal{T})\setminus\mathcal{A},k} & k\in\mathcal{L}\setminus\mathcal{T}\\ \mathsf{Ind}_{\mathcal{A}\setminus\mathcal{T},k} & k\in\mathcal{A}\setminus\mathcal{T}\end{cases},\quad(17b)$$

$$\forall\mathcal{T}\in\Omega^{t}_{\mathcal{A}\cup\mathcal{L}},\ \forall k\in(\mathcal{A}\cup\mathcal{L})\setminus\mathcal{T},\quad(17c)$$

for some constants $\{\mathsf{c}^{(\mathcal{A})}_{\mathcal{T}}\in\mathbb{F}_{\mathsf{q}}:\mathcal{T}\in\Omega^{t}_{\mathcal{A}\cup\mathcal{L}},\ \forall k\in(\mathcal{A}\cup\mathcal{L})\setminus\mathcal{T}\}$. Finally, the relationships in (17) can be represented on an undirected graph that has the $\widetilde{\beta}^{(\mathcal{A})}_{\mathcal{S}}$'s and the $\mathsf{c}^{(\mathcal{A})}_{\mathcal{T}}$'s as vertices and whose edges are labeled by the encoding coefficients according to the constraints in (17a). *A spanning tree on such a graph identifies all the encoding coefficients that are free to vary*, in other words, cycles on such a graph identify constraints that the encoding coefficients must satisfy.

Remark. The reason why the signs of the encoding coefficients (and the resulting decoding coefficients) must alternate in [5] is because of the condition in (17b), which is satisfied by the choice in (14); however the alternating patten in (14) in just one possible feasible linear scheme. The choice of coefficients in (14) (and the resulting decoding coefficients) has the following advantages: (a) the scheme does not involve divisions other than by elements of unit modulo, which in turns allows one to extend the scheme to monomial retrieval as well [5]; and (b) the scheme works irrespective of the characteristics of the finite field. □

IV. Proof of Main Result

We prove here the result in Section III for the case $\mathsf{K}-|\mathcal{L}|=t+1$ (i.e., only the multicast message indexed by $\mathcal{A}=\overline{\mathcal{L}}$ must be reconstructed in (12)). The case $\mathsf{K}-|\mathcal{L}|>t+1$ can be solved by analyzing several systems with only $|\mathcal{L}|+t+1$ users each based on the result proved here; the proof however is not reported here for sake of space and may be found in [1]. Moreover, we provide a complete characterization of all feasible linear schemes via graph theoretic properties. The proof holds for all $r=|\mathcal{L}|\in[\min(\mathsf{K},\mathsf{N})]$ and $t\in[0:\mathsf{K}]$.

We consider here a system with K users, $r=|\mathcal{L}|$ leaders, and memory size parameterized by t, where (t,r) are fixed and satisfy $\mathsf{K}=r+t+1$. For a subset $\mathcal{T}$ of $[\mathsf{K}]$, we let $\overline{\mathcal{T}}:=[\mathsf{K}]\setminus\mathcal{T}$. In particular, $\overline{\mathcal{L}}$ is the set of non-leader users.

Define the transformed demand matrix as in (7). Only the multicast message indexed by $\mathcal{A}=\overline{\mathcal{L}}$ needs to be reconstructed, thus for notation convenience we drop $\mathcal{A}$ from $\beta^{(\mathcal{A})}_{\mathcal{S}}$ in (12). We re-write (12) with $\beta_{\overline{\mathcal{L}}}=-1$ (but actually any non-zero value will do), as follow

$$\mathbb{F}_{\mathsf{q}}^{\mathsf{B}/\binom{\mathsf{K}}{t}}\ni 0=\sum_{\mathcal{S}\in\Omega^{t+1}_{[\mathsf{K}]}}\beta_{\mathcal{S}}W_{\mathcal{S}}\quad(18a)$$

$$=\sum_{\mathcal{S}\in\Omega^{t+1}_{[\mathsf{K}]}}\beta_{\mathcal{S}}\sum_{k\in\mathcal{S}}\alpha_{k,\mathcal{S}\setminus\{k\}}\sum_{\ell\in\mathcal{L}}[\mathbb{D}']_{k,\ell}B_{\ell,\mathcal{S}\setminus\{k\}}\quad(18b)$$

$$=\sum_{\mathcal{T}\in\Omega^{t}_{[K]}}\sum_{\ell\in\mathcal{L}}\sum_{k\in\overline{\mathcal{T}}}\beta_{\{k\}\cup\mathcal{T}}\ \alpha_{k,\mathcal{T}}\ [\mathbb{D}']_{k,\ell}B_{\ell,\mathcal{T}}.\quad(18c)$$

Since (18) must hold for all $\{B_{\ell,\mathcal{T}}\in\mathbb{F}_{\mathsf{q}}^{\mathsf{B}/\binom{\mathsf{K}}{t}}:\ell\in\mathcal{L},\ \mathcal{T}\in\Omega^{t}_{[K]}\}$, we equivalently rewrite it, $\forall\ell\in\mathcal{L},\ \forall\mathcal{T}\in\Omega^{t}_{[K]}$, as

$$\mathbb{F}_{\mathsf{q}}\ni 0=\sum_{k\in\overline{\mathcal{T}}}\beta_{\{k\}\cup\mathcal{T}}\ \alpha_{k,\mathcal{T}}\ [\mathbb{D}']_{k,\ell}\quad(19a)$$

$$=\sum_{k\in\overline{\mathcal{T}}\cap\mathcal{L}}\beta_{\{k\}\cup\mathcal{T}}\ \alpha_{k,\mathcal{T}}\ 1_{\{k=\ell\}}\quad(19b)$$

$$+\sum_{k\in\overline{\mathcal{T}}\cap\overline{\mathcal{L}}}\beta_{\{k\}\cup\mathcal{T}}\ \alpha_{k,\mathcal{T}}\ x_{k,\ell},\quad(19c)$$

by the dentition of transformed demand matrix in (7). We finally rewrite (19) by separating it into two cases

$$\sum_{k\in\overline{\mathcal{T}}\cap\overline{\mathcal{L}}}\beta_{\{k\}\cup\mathcal{T}}\ \alpha_{k,\mathcal{T}}\ x_{k,\ell}=\begin{cases}0 & \ell\in\mathcal{L}\cap\mathcal{T},\\ -\beta_{\{\ell\}\cup\mathcal{T}}\ \alpha_{\ell,\mathcal{T}} & \ell\in\mathcal{L}\cap\overline{\mathcal{T}}\end{cases},$$

$$\forall\mathcal{T}\in\Omega^{t}_{[K]}.\quad(20)$$

Next, we say that a set $\mathcal{T}\subseteq[\mathsf{K}]$ is in 'hierarchy h' if $|\mathcal{T}\cap\mathcal{L}|=h$ for some $h\in[0:\min(|\mathcal{T}|,|\mathcal{L}|)]$. We also say that $\beta_{\mathcal{S}}$ is in hierarchy h if $\mathcal{S}$ is in hierarchy h. *We next seek to show that in general the decoding coefficients in hierarchy $h+1$ can be expressed as a linear combination of those in hierarchy h.*

Initialization / hierarchy $h=1$: $\beta_{\overline{\mathcal{L}}}=-1$ is the only decoding coefficient in hierarchy 0. By picking $\mathcal{T}=\overline{\mathcal{L}}\setminus\{u\}$, $u\in\overline{\mathcal{L}}$, and $\ell\in\mathcal{L}$ in (20) (and thus $\overline{\mathcal{T}}\cap\overline{\mathcal{L}}=\{u\}$), we express the decoding coefficients in hierarchy 1 as follows

$$\beta_{\{\ell\}\cup\overline{\mathcal{L}}\setminus\{u\}}=\frac{\alpha_{u,\overline{\mathcal{L}}\setminus\{u\}}}{\alpha_{\ell,\overline{\mathcal{L}}\setminus\{u\}}}\ x_{u,\ell},\ \forall u\in\overline{\mathcal{L}},\ \forall\ell\in\mathcal{L}.\quad(21)$$

Hierarchy h: For any $\mathcal{T}\in\Omega^{t}_{[K]}$, from (20) with $\ell\in\mathcal{T}$,

$$\sum_{k\in\overline{\mathcal{T}}\cap\overline{\mathcal{L}}}\beta_{\{k\}\cup\mathcal{T}}\ \alpha_{k,\mathcal{T}}\ x_{k,\ell}=0,\ \forall\ell\in\mathcal{L}\cap\mathcal{T}.\quad(22)$$

In particular, for a $\mathcal{T}$ in hierarchy $h>0$, we indicate WLOG (recall that here $|\overline{\mathcal{L}}|=\mathsf{K}-r=t+1=|\mathcal{T}|+1$ and thus $|\mathcal{T}\cap\mathcal{L}|=h$, $|\mathcal{T}\cap\overline{\mathcal{L}}|=t-h$, $|\overline{\mathcal{T}}\cap\mathcal{L}|=r-h$, $|\overline{\mathcal{T}}\cap\overline{\mathcal{L}}|=h+1$)

$$\mathcal{T}\cap\mathcal{L}=\{\ell_1,\ldots,\ell_h\}:\ell_1<\ldots<\ell_h,\ \text{(leaders)},\quad(23)$$

$$\overline{\mathcal{T}} \cap \overline{\mathcal{L}} = \{j_1, \dots, j_h, j_{h+1}\} : j_1 < \dots < j_{h+1}, \tag{24}$$

and collect the h constraints in (22) in matrix form as indicated in (25) and (26), at the top of the next page, for all $\mathcal{T} \in \Omega^t_{[K]}$. By Cramer's rule, the solution of (26) can be written as

$$(-1)^{h+1-i} \frac{\det\left(\mathbb{D}'[\overline{\mathcal{T}} \cap \overline{\mathcal{L}} \setminus \{j_i\}, \mathcal{L} \cap \mathcal{T}]\right)}{\det\left(\mathbb{D}'[\overline{\mathcal{T}} \cap \overline{\mathcal{L}} \setminus \{j_{h+1}\}, \mathcal{L} \cap \mathcal{T}]\right)} \tag{27a}$$

$$= \frac{\beta_{\{j_i\} \cup \mathcal{T}}\ \alpha_{j_i,\mathcal{T}}}{\beta_{\{j_{h+1}\} \cup \mathcal{T}}\ \alpha_{j_{h+1},\mathcal{T}}}, \forall i \in [h], \forall j_i \in \overline{\mathcal{T}} \cap \overline{\mathcal{L}}, \tag{27b}$$

or equivalently (27) can be written as (recall $j \in \overline{\mathcal{T}} \cap \overline{\mathcal{L}}$)

$$(-1)^1 \frac{\beta_{\{j_1\} \cup \mathcal{T}}\ \alpha_{j_1,\mathcal{T}}}{\det\left(\mathbb{D}'[\overline{\mathcal{T}} \cap \overline{\mathcal{L}} \setminus \{j_1\}, \mathcal{L} \cap \mathcal{T}]\right)} = \dots \tag{28a}$$

$$= (-1)^{h+1} \frac{\beta_{\{j_{h+1}\} \cup \mathcal{T}}\ \alpha_{j_{h+1},\mathcal{T}}}{\det\left(\mathbb{D}'[\overline{\mathcal{T}} \cap \overline{\mathcal{L}} \setminus \{j_{h+1}\}, \mathcal{L} \cap \mathcal{T}]\right)}. \tag{28b}$$

Notice that all the decoding coefficients in (28) are in hierarchy h if the set $\mathcal{T}$ is hierarchy h.

Hierarchy $h+1$: We plug the decoding coefficients in hierarchy h from (28) into (20) with $\ell \in \overline{\mathcal{T}}$ and, by definition of determinant (i.e., Laplace expansion along a column), we obtain that for all $\mathcal{T} \in \Omega^t_{[K]}$

$$-\beta_{\{\ell\} \cup \mathcal{T}}\ \alpha_{\ell,\mathcal{T}} = \sum_{k \in \overline{\mathcal{T}} \cap \overline{\mathcal{L}}} \beta_{\{k\} \cup \mathcal{T}}\ \alpha_{k,\mathcal{T}}\ x_{k,\ell} \tag{29a}$$

$$= \frac{\beta_{\{j_{h+1}\} \cup \mathcal{T}}\ \alpha_{j_{h+1},\mathcal{T}}}{\det\left(\mathbb{D}'[\overline{\mathcal{T}} \cap \overline{\mathcal{L}} \setminus \{j_{h+1}\}, \mathcal{L} \cap \mathcal{T}]\right)} \tag{29b}$$

$$\cdot \sum_{i \in [h+1]} (-1)^{h+1-i} \det\left(\mathbb{D}'[\overline{\mathcal{T}} \cap \overline{\mathcal{L}} \setminus \{j_i\}, \mathcal{L} \cap \mathcal{T}]\right)\ x_{j_i,\ell}$$

$$= (-1)^{h+1} \frac{\beta_{\{j_{h+1}\} \cup \mathcal{T}}\ \alpha_{j_{h+1},\mathcal{T}}}{\det\left(\mathbb{D}'[\overline{\mathcal{T}} \cap \overline{\mathcal{L}} \setminus \{j_{h+1}\}, \mathcal{L} \cap \mathcal{T}]\right)} \tag{29c}$$

$$\cdot (-1)^{-\mathsf{Ind}_{\mathcal{L} \cap \mathcal{T} \cup \{\ell\},\ell}} \det\left(\mathbb{D}'[\overline{\mathcal{T}} \cap \overline{\mathcal{L}}, \mathcal{L} \cap \mathcal{T} \cup \{\ell\}]\right), \tag{29d}$$

or equivalently, $\forall \mathcal{T} \in \Omega^t_{[K]}, \forall \ell \in \overline{\mathcal{T}} \cap \mathcal{L}$, we have

$$(-1)^{1+\mathsf{Ind}_{\mathcal{L} \cap \mathcal{T} \cup \{\ell\},\ell}} \frac{\beta_{\{\ell\} \cup \mathcal{T}}\ \alpha_{\ell,\mathcal{T}}}{\det\left(\mathbb{D}'[\overline{\mathcal{T}} \cap \overline{\mathcal{L}}, \mathcal{L} \cap \mathcal{T} \cup \{\ell\}]\right)} = \text{eq(28)}. \tag{30}$$

Notice that all the decoding coefficients in (30) are in hierarchy $h+1$ if the set $\mathcal{T}$ is hierarchy h.

Combing everything together: We can interpret (28) and (30) as follows: for a set $\mathcal{T} \in \Omega^t_{[K]}$ and an element $k \in \overline{\mathcal{T}}$, we create a set $\mathcal{S} = \mathcal{T} \cup \{k\} \in \Omega^{t+1}_{[K]}$ that satisfies the following: add a non-leader

$$k = j \in \overline{\mathcal{T}} \cap \overline{\mathcal{L}} : \overline{\mathcal{T}} \cap \overline{\mathcal{L}} \setminus \{j\} = \overline{\mathcal{L}} \setminus (\{j\} \cup \mathcal{T}), \tag{31a}$$

$$\mathcal{L} \cap \mathcal{T} = (\{j\} \cup \mathcal{T}) \setminus \overline{\mathcal{L}}, \tag{31b}$$

or add a leader

$$k = \ell \in \overline{\mathcal{T}} \cap \mathcal{L} : \overline{\mathcal{T}} \cap \overline{\mathcal{L}} = \overline{\mathcal{L}} \setminus (\{\ell\} \cup \mathcal{T}), \tag{31c}$$

$$\mathcal{L} \cap \mathcal{T} \cup \{\ell\} = (\{\ell\} \cup \mathcal{T}) \setminus \overline{\mathcal{L}}, \tag{31d}$$

thus (recall $\overline{\mathcal{T}} \cap \overline{\mathcal{L}} = \overline{\mathcal{L}} \setminus \mathcal{T}$, $\overline{\mathcal{T}} \cap \mathcal{L} = \mathcal{L} \setminus \mathcal{T}$ and $\overline{\mathcal{T}} = [\mathsf{K}] \setminus \mathcal{T}$)

$$\mathsf{c}^{(\overline{\mathcal{L}})}_{\mathcal{T}} = (-1)^{\phi^{(\overline{\mathcal{L}})}_{k,\mathcal{T}}} \alpha_{k,\mathcal{T}} \cdot \widetilde{\beta}^{(\overline{\mathcal{L}})}_{\{k\} \cup \mathcal{T}}, \forall \mathcal{T} \in \Omega^t_{[K]},\ \forall k \in \overline{\mathcal{T}}, \tag{32a}$$

$$\widetilde{\beta}^{(\overline{\mathcal{L}})}_{\{k\} \cup \mathcal{T}} := \frac{\beta_{\{k\} \cup \mathcal{T}}}{\det\left(\mathbb{D}'[\overline{\mathcal{L}} \setminus (\{k\} \cup \mathcal{T}), (\{k\} \cup \mathcal{T}) \setminus \overline{\mathcal{L}}]\right)}, \tag{32b}$$

$$\phi^{(\overline{\mathcal{L}})}_{k,\mathcal{T}} := \begin{cases} \mathsf{Ind}_{\overline{\mathcal{L}} \setminus \mathcal{T},k} & k \in \overline{\mathcal{L}} \setminus \mathcal{T}, \\ 1 + \mathsf{Ind}_{(\{k\} \cup \mathcal{T}) \setminus \overline{\mathcal{L}},k} & k \in \mathcal{L} \setminus \mathcal{T}, \end{cases} \tag{32c}$$

for some constatns $\{\mathsf{c}^{(\overline{\mathcal{L}})}_{\mathcal{T}} : \mathcal{T} \in \Omega^t_{[K]}\}$.

The term in (32b) (that only depends on $\{k\} \cup \mathcal{T}$ as opposed to on both k and $\mathcal{T}$) can be further expressed as a function of the encoding coefficients as follows. For a set $\mathcal{S} \in \Omega^{t+1}_{[\mathsf{K}]}, \mathcal{S} \neq \overline{\mathcal{L}}$, in hierarchy h and by setting WLOG

$$\mathcal{S} \cap \mathcal{L} = \{\ell_1, \dots, \ell_h\} : \ell_1 < \dots < \ell_h, \text{ (leaders)} \tag{33}$$

$$\overline{\mathcal{S}} \cap \overline{\mathcal{L}} = \{j_1, \dots, j_h\} : j_1 < \dots < j_h, \text{ (non leaders)} \tag{34}$$

$$\mathcal{S} \cap \overline{\mathcal{L}} = \mathcal{J},\ \overline{\mathcal{L}} = \{j_1, \dots, j_h\} \cup \mathcal{J}, \tag{35}$$

we iteratively use (29) to express $\beta_{\mathcal{S}}$ with $\mathcal{S} = \{\ell_1 \dots \ell_h\} \cup \mathcal{J}$ as in (36) at the top of the next page and where the last equality follows since by definition $\beta_{\{j_h \dots j_1\} \cup \mathcal{J}} = \beta_{\overline{\mathcal{L}}} = -1$ and by convention $\det(\mathbb{D}'[\emptyset, \emptyset]) = 1$. Eq (36) shows that each decoding coefficient is proportional to the determinant of a sub-matrix of the transformed demand matrix and that the proportionality coefficient (denoted as $\widetilde{\beta}^{(\overline{\mathcal{L}})}_{\{\ell_1 \dots \ell_h\} \cup \mathcal{J}}$) depends only on the encoding coefficients; the encoding coefficients however are not all free to vary, as they need to satisfy the relationships imposed by (32a).

Graph representation: The relationships among $\mathcal{V}_1 := \{\mathsf{c}^{(\overline{\mathcal{L}})}_{\mathcal{T}} : \mathcal{T} \in \Omega^t_{[K]}\}$ and $\mathcal{V}_2 := \{\widetilde{\beta}^{(\overline{\mathcal{L}})}_{\mathcal{S}} : \mathcal{S} \in \Omega^{t+1}_{[K]}\}$ imposed by (32) can be represented by a graph. We create an undirected graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$, where $\mathcal{V} := \mathcal{V}_1 \cup \mathcal{V}_2$ is the vertex set and $\mathcal{E} := \{(\widetilde{\beta}^{(\overline{\mathcal{L}})}_{\{k\} \cup \mathcal{T}}, \mathsf{c}^{(\overline{\mathcal{L}})}_{\mathcal{T}}) : \mathcal{T} \in \Omega^t_{[K]}, k \in \overline{\mathcal{T}}\}$ is the edge set. We assign label $(-1)^{\phi^{(\overline{\mathcal{L}})}_{k,\mathcal{T}}} \alpha_{k,\mathcal{T}}$ to edge $(\widetilde{\beta}^{(\overline{\mathcal{L}})}_{\{k\} \cup \mathcal{T}}, \mathsf{c}^{(\overline{\mathcal{L}})}_{\mathcal{T}}) \in \mathcal{E}$ to capture the relationship in (32). We elect $\widetilde{\beta}_{\overline{\mathcal{L}}}$ to be the root node and assign to it the value -1 (but we could start from any other vertex with any non-zero value). We then create a spanning tree from that root[2]. By doing so, we find values for all the vertices by using (32). One can easily see, by the properties of spanning trees, that the encoding coefficients on the edges of the spanning tree are free to vary (i.e., they can be be any non-zero value), while the encoding coefficients on edges that are not part of the spanning tree are determined through the following relationship: every path from the root to a node determines the value of the node by using (32) and all those values must be equal; in other words, every cycle in the graph, obtained by adding a edge that is not on the spanning tree to the spanning tree, is a constraint.

This concludes the proof for the case $\mathsf{K} - r = t + 1$.

Example: Fig. 1 shows the described graph for the case of $K = 4$ users, $r = 2$ leaders, and memory size $t = 1$ (i.e., each user can cache one file); the edges of a possible spanning tree are marked by a solid red line; the edges that are not in

[2]A spanning tree is a subset of the graph, which has *all the vertices of the graph covered with minimum possible number of edges*. Hence, a spanning tree does not have cycles and it cannot be disconnected. Moreover, every connected and undirected graph has at least one spanning tree.

$$\begin{bmatrix}\beta_{\{j_1\}\cup\mathcal{T}}\ \alpha_{j_1,\mathcal{T}} & \cdots & \beta_{\{j_h\}\cup\mathcal{T}}\ \alpha_{j_h,\mathcal{T}} & \beta_{\{j_{h+1}\}\cup\mathcal{T}}\ \alpha_{j_{h+1},\mathcal{T}}\end{bmatrix} \underbrace{\begin{bmatrix} x_{j_1,\ell_1} & \cdots & x_{j_1,\ell_h} \\ \vdots & \ddots & \vdots \\ x_{j_h,\ell_1} & \cdots & x_{j_h,\ell_h} \\ x_{j_{h+1},\ell_1} & \cdots & x_{j_{h+1},\ell_h}\end{bmatrix}}_{=\mathbb{D}'[\overline{\mathcal{T}}\cap\overline{\mathcal{L}},\mathcal{L}\cap\mathcal{T}]\in\mathbb{F}_{\mathsf{q}}^{h+1\times h}} = 0 \in \mathbb{F}_{\mathsf{q}}^{1\times h}, \tag{25}$$

$$\begin{bmatrix}\frac{\beta_{\{j_1\}\cup\mathcal{T}}\ \alpha_{j_1,\mathcal{T}}}{\beta_{\{j_{h+1}\}\cup\mathcal{T}}\ \alpha_{j_{h+1},\mathcal{T}}} & \cdots & \frac{\beta_{\{j_h\}\cup\mathcal{T}}\ \alpha_{j_h,\mathcal{T}}}{\beta_{\{j_{h+1}\}\cup\mathcal{T}}\ \alpha_{j_{h+1},\mathcal{T}}}\end{bmatrix} \underbrace{\begin{bmatrix} x_{j_1,\ell_1} & \cdots & x_{j_1,\ell_h} \\ \vdots & \ddots & \vdots \\ x_{j_h,\ell_1} & \cdots & x_{j_h,\ell_h}\end{bmatrix}}_{=\mathbb{D}'[\overline{\mathcal{T}}\cap\overline{\mathcal{L}}\setminus\{j_{h+1}\},\mathcal{L}\cap\mathcal{T}]\in\mathbb{F}_{\mathsf{q}}^{h\times h}} = -\underbrace{\begin{bmatrix} x_{j_{h+1},\ell_1} & \cdots & x_{j_{h+1},\ell_h}\end{bmatrix}}_{=\mathbb{D}'[\{j_{h+1}\},\mathcal{L}\cap\mathcal{T}]\in\mathbb{F}_{\mathsf{q}}^{1\times h}}, \tag{26}$$

$$\widetilde{\beta}^{(\overline{\mathcal{L}})}_{\{\ell_1\ldots\ell_h\}\cup\mathcal{J}} = \frac{\beta_{\{\ell_1\ldots\ell_h\}\cup\mathcal{J}}}{\det\left(\mathbb{D}'[\overline{\mathcal{S}}\cap\overline{\mathcal{L}},\mathcal{S}\cap\mathcal{L}]\right)} = -\frac{\alpha_{j_h,\{\ell_1\ldots\ell_{h-1}\}\cup\mathcal{J}}}{\alpha_{\ell_h,\{\ell_1\ldots\ell_{h-1}\}\cup\mathcal{J}}}\frac{\beta_{\{j_h\}\cup\{\ell_1\ldots\ell_{h-1}\}\cup\mathcal{J}}}{\det\left(\mathbb{D}'[\overline{\mathcal{S}}\cap\overline{\mathcal{L}}\setminus\{j_h\},\mathcal{S}\cap\mathcal{L}\setminus\{\ell_h\}]\right)} \tag{36a}$$

$$= (-1)^h\ \frac{\alpha_{j_h,\{\ell_1\ldots\ell_{h-1}\}\cup\mathcal{J}}}{\alpha_{\ell_h,\{\ell_1\ldots\ell_{h-1}\}\cup\mathcal{J}}}\ \frac{\alpha_{j_{h-1},\{j_h\}\cup\{\ell_1\ldots\ell_{h-2}\}\cup\mathcal{J}}}{\alpha_{\ell_{h-1},\{j_h\}\cup\{\ell_1\ldots\ell_{h-2}\}\cup\mathcal{J}}}\cdots\frac{\alpha_{j_1,\{j_h\ldots j_2\}\cup\mathcal{J}}}{\alpha_{\ell_1,\{j_h\ldots j_2\}\cup\mathcal{J}}}\ \frac{\beta_{\{j_h\ldots j_1\}\cup\mathcal{J}}}{\det\left(\mathbb{D}'[\emptyset,\emptyset]\right)} \tag{36b}$$

$$= (-1)^{h+1}\ \prod_{i=1}^{h}\frac{\alpha_{j_i,\{j_h\ldots j_{i+1}\}\cup\{\ell_1\ldots\ell_{i-1}\}\cup\mathcal{J}}}{\alpha_{\ell_i,\{j_h\ldots j_{i+1}\}\cup\{\ell_1\ldots\ell_{i-1}\}\cup\mathcal{J}}}, \tag{36c}$$

the spanning tree (doted blue line edges) correspond to the following constraints

$$\text{vertex } \mathsf{c}_1 : \alpha_{3,\{1\}} = -\alpha_{1,\{3\}}\frac{\alpha_{4,\{1\}}}{\alpha_{1,\{4\}}}\frac{\alpha_{3,\{4\}}}{\alpha_{4,\{3\}}}, \tag{37a}$$

$$\text{vertex } \mathsf{c}_2 : \alpha_{3,\{2\}} = -\alpha_{2,\{3\}}\frac{\alpha_{4,\{2\}}}{\alpha_{2,\{4\}}}\frac{\alpha_{3,\{4\}}}{\alpha_{4,\{3\}}}, \tag{37b}$$

$$\text{vertex } \widetilde{\beta}_{\{1,2\}} : \alpha_{2,\{1\}} = -\frac{\alpha_{4,\{1\}}}{\alpha_{1,\{4\}}}\frac{\alpha_{4,\{2\}}}{\alpha_{2,\{4\}}}\alpha_{1,\{2\}}. \tag{37c}$$

The relationships in (37) can arrived at by directly solving (12) as shown in the appendix of [1].

V. Conclusion

In this paper, we investigated the constraints that a linear scheme for cache-aided scalar linear function retrieval must satisfy in order to be feasible. We showed that the constraints among the parameters of a feasible linear scheme are captured by the cycles of a certain graph. Equivalently, we showed that a spanning tree for the graph identifies all the parameters of the scheme that are free to vary. The structure of our general scheme sheds light into a scheme that had been previously proposed in the literature. Ongoing work includes using similar ideas to possibly extend the scheme in [6].

Acknowedgment

This work was supported in part by NSF Award 1910309.

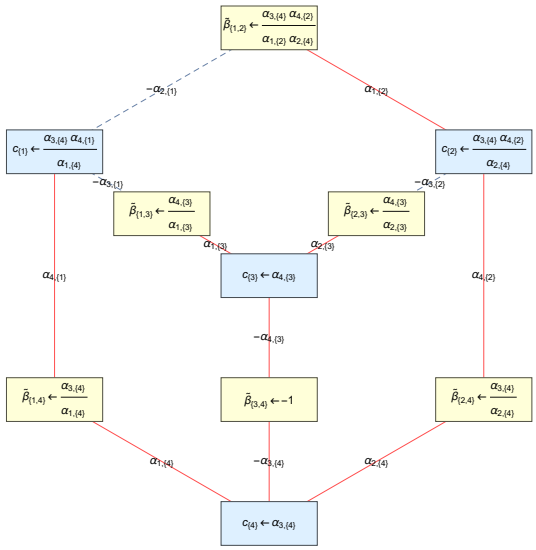

Fig. 1: The graph and a possible spanning tree for the case $\mathsf{K} = 4, r = 2, t = 1$. For legibility, we removed the superscript $\overline{\mathcal{L}} = \{3,4\}$ from the vertices. The edges are labeled by an encoding coefficient with an appropriate sign. Solid edges form a spanning tree; the encoding coefficients on dotted edges are determined by using (32a). The β-vertexes are in a yellow box and the c-vertexes in a cyan box; the expression on the RHS of the symbol $\leftarrow$ in a box is the value assigned to the vertex when we travel the graph from the root (i.e., $\widetilde{\beta}_{\{3,4\}} = -1$) along the spanning tree.

REFERENCES

[1] Y. Ma and D. Tuninetti, "A general coded caching scheme for scalar linear function retrieval," *arXiv preprint arXiv:2102.02122*, 2021.

[2] M. A. Maddah-Ali and U. Niesen, "Fundamental limits of caching," *IEEE Transactions on Information Theory*, vol. 60, no. 5, pp. 2856–2867, 2014.

[3] Q. Yu, M. A. Maddah-Ali, and A. S. Avestimehr, "The exact rate-memory tradeoff for caching with uncoded prefetching," *IEEE Transactions on Information Theory*, vol. 64, no. 2, pp. 1281–1296, 2017.

[4] K. Wan, D. Tuninetti, and P. Piantanida, "An index coding approach to caching with uncoded cache placement," *IEEE Transactions on Information Theory*, vol. 66, no. 3, pp. 1318–1332, 2020.

[5] K. Wan, H. Sun, M. Ji, D. Tuninetti, and G. Caire, "Cache-aided scalar linear function retrieval," in *2020 IEEE International Symposium on Information Theory (ISIT)*, pp. 1717–1722, IEEE, 2020.

[6] H. Sun and S. A. Jafar, "The capacity of private computation," *IEEE Transactions on Information Theory*, vol. 65, no. 6, pp. 3880–3897, 2018.

[7] Q. Yu, M. A. Maddah-Ali, and A. S. Avestimehr, "Characterizing the rate-memory tradeoff in cache networks within a factor of 2," *IEEE Transactions on Information Theory*, vol. 65, no. 1, pp. 647–663, 2018.