entropy

MDPI

Article

An Umbrella Converse for Data Exchange: Applied to Caching, Computing, and Shuffling [†]

Prasad Krishnan [1,*], Lakshmi Natarajan [2] and V. Lalitha [1]

[1] Signal Processing & Communications Research Center, International Institute of Information Technology Hyderabad, Hyderabad 500032, India; lalitha.v@iiit.ac.in
[2] Department of Electrical Engineering, Indian Institute of Technology Hyderabad, Kandi 502205, India; lakshminatarajan@iith.ac.in
* Correspondence: prasad.krishnan@iiit.ac.in

[†] Part of this work was presented at the IEEE Information Theory Workshop 2020 held virtually from 11–15 April 2021.

Abstract: The problem of data exchange between multiple nodes with storage and communication capabilities models several current multi-user communication problems like Coded Caching, Data Shuffling, Coded Computing, etc. The goal in such problems is to design communication schemes which accomplish the desired data exchange between the nodes with the optimal (minimum) amount of communication load. In this work, we present a converse to such a general data exchange problem. The expression of the converse depends only on the number of bits to be moved between different subsets of nodes, and does not assume anything further specific about the parameters in the problem. Specific problem formulations, such as those in Coded Caching, Coded Data Shuffling, and Coded Distributed Computing, can be seen as instances of this generic data exchange problem. Applying our generic converse, we can efficiently recover known important converses in these formulations. Further, for a generic coded caching problem with heterogeneous cache sizes at the clients with or without a central server, we obtain a new general converse, which subsumes some existing results. Finally we relate a "centralized" version of our bound to the known generalized independence number bound in index coding and discuss our bound's tightness in this context.

Keywords: data exchange; coded caching; coded distributed computing; coded data shuffling; converse; index coding

check for updates

Citation: Krishnan, P.; Natarajan, L.; Lalitha, V. An Umbrella Converse for Data Exchange: Applied to Caching, Computing, and Shuffling. *Entropy* **2021**, *23*, 985. https://doi.org/10.3390/e23080985

Academic Editors: Siu-Wai Ho, Lawrence Ong and Kenneth Shum

Received: 23 May 2021
Accepted: 24 July 2021
Published: 30 July 2021

1. Introduction and Main Result

Consider a system of K *nodes*, denoted by $[K] \triangleq \{1, \ldots, K\}$, each of which have (not necessarily uniform) storage. The nodes can communicate with each other through a noiseless bus link, in which transmissions of any node is received by all others. Each node possesses a collection of data symbols (represented in bits) in its local storage and demands another set of symbols present in other nodes. We formalize this as a *data exchange problem*.

Definition 1. *A data exchange problem on a set of K nodes involving a collection B of information bits is given by the following:*

- *a collection $\{C_i : i \in [K]\}$, where $C_i \subset B$ denotes the subset of data present in node i,*
- *a collection $\{D_i : i \in [K]\}$ where $D_i \subset \cup_{j \neq i} C_j \setminus C_i$ denotes the set of bits demanded by node i.*

The above data exchange problem models a number of cache-enabled multi-receiver communication problems studied recently in the coding theory community, including Coded Caching [1], Coded Distributed Computing [2,3], Coded Data Shuffling [4–6], and Coded Data Rebalancing [7]. In [8], a special case of our general problem here was considered in the name of *cooperative data exchange*, where the goal was to reach a state in which all nodes have all the data in the system.

A solution to a given data exchange problem involves communication between the nodes. Each node i encodes the symbols in C_i into a codeword of length l_i and sends it to all other nodes. The respective demanded symbols at any node is then to be decoded using the received transmissions from all the other nodes and the node's own content.

Formally, a *communication scheme* for the given data exchange problem consists of a set of encoding functions $\Phi \triangleq \{\phi_i : i \in [K]\}$ and decoding functions $\Psi \triangleq \{\psi_i : i \in [K]\}$, defined as follows.

$$\begin{aligned}\phi_i :& \{0,1\}^{|C_i|} \rightarrow \{0,1\}^{l_i}, \text{(for some non-negative integer } l_i\text{)}\\ \psi_i :& \{0,1\}^{|C_i|} \times \{0,1\}^{\sum\limits_{j\neq i} l_j} \rightarrow \{0,1\}^{|D_i|},\end{aligned}$$

such that

$$\psi_i\big(C_i, \{\phi_j(C_j) : j \neq i\}\big) = D_i.$$

The *communication load* of the above scheme is defined as the total number of bits communicated, i.e.,

$$L(\Phi, \Psi) \triangleq \sum_{i\in[K]} l_i.$$

The optimal communication load is then denoted by

$$L^* \triangleq \min_{\Phi,\Psi} L(\Phi, \Psi).$$

The central result in this work is Theorem 1 in Section 1.1, which is a lower bound on the optimal communication load L^*. Using this lower bound, we recover several important converse results of cache-enabled communication problems studied in the literature, including Coded Caching (Section 2), Data Shuffling (Section 3), and Distributed Computing (Section 4). In each of these sections, we briefly review each setting and then apply Theorem 1 to recover the respective converses. As a result, the proofs of these existing converses are also made simpler than what is already available in the literature for the respective settings. The generic structure of the converse proofs obtained using our data exchange bound is presented in Section 1.2. This structure includes three steps, which we also highlight at the appropriate junctures within the proofs themselves. The close relationship between these problems is quite widely known. This work gives a further formal grounding to this connection, by abstracting the common structure of these converses into a general form, which can potentially be applied to other new data exchange problems as well.

Apart from recovering existing results, more importantly we also use our data exchange lower bound to obtain *new* tight converse results for some settings, while improving tightness results of some known bounds. Specifically, we present a new converse for a generic coded caching setting with multi-level cache sizes. Using this, we are able to close the gap to optimality for some known special cases of this generic setting (Section 2.1). In Section 5, we show the relationship between a "centralized" version of our data exchange lower bound and an existing bound for index coding known as the α-bound or the generalized independence number bound [9]. In general, we find that our bound is weaker than the α-bound. However, for unicast index coding problems, we identify the precise conditions under which our data exchange bound is equal to the α-bound. In Section 6, we discuss the application of our data exchange lower bound to more generalized index coding settings, specifically distributed index coding [10,11] and embedded index coding [12].

Notation: For positive integer a, let $[a] \triangleq \{1, \ldots, a\}$. For a set S, we denote by $S \setminus k$ the set of items in S except for the item k, and represent the union $S \cup \{k\}$ as $S \cup k$. The binomial coefficient is denoted by $\binom{n}{k}$, which is zero if $k > n$. The set of all t-sized subsets of a set A is denoted by $\binom{A}{t}$.

1.1. A Converse for the Data Exchange Problem

In this subsection, we will obtain a lower bound on the optimal communication load of the general data exchange problem defined in Section 1. This is the central result of this work. The predecessor to the proof technique of our data exchange lower bound is in [3], which first presented an induction based approach for the converse of the coded distributed computing setting. Our proof uses a similar induction technique.

Given a data exchange problem and for $P, Q \subset [K]$ such that $P \neq \emptyset$, let a_P^Q denote the number of bits which are stored in every node in the subset of nodes Q and stored in no other node, and demanded by every node in the subset P and demanded by no other node, i.e.,

$$a_P^Q \triangleq |(\cap_{i\in P} D_i) \cap (\cap_{j\in Q} C_j) \setminus \Big((\cup_{j\notin Q} C_j) \cup (\cup_{i\notin P} D_i) \Big)|. \quad (1)$$

Note that, by definition, $a_P^Q = 0$ under the following conditions.

- If $P \cap Q \neq \emptyset$, as the bits demanded by any node are absent in the same node.
- If $Q = \emptyset$, by Definition 1.

Theorem 1 gives a lower bound on the optimal communication load of a given data exchange problem. The proof of the theorem is relegated to Appendix A. The idea of the proof is as follows. If we consider only two nodes in the system, say $[K] = \{1, 2\}$, then each of the 2 nodes has to transmit whatever bits it has which are demanded by the other node, i.e., $L^* \geq a_{\{2\}}^{\{1\}} + a_{\{1\}}^{\{2\}}$. The proof of the theorem uses this as a base case and employs an induction technique to obtain a sequence of cut-set bounds leading to the final expression.

Theorem 1.

$$L^* \geq \sum_{P\subset[K]} \sum_{Q\subset[K]\setminus P} \frac{|P|}{|P| + |Q| - 1} a_P^Q.$$

Theorem 1, along with the observation that $a_\emptyset^Q = 0 = a_P^\emptyset$ gives us the following corollary, which is a restatement of Theorem 1.

Corollary 1. *Let*

$$n(p, q) \triangleq \sum_{\substack{P,Q\subset[K]: \\ |P|=p,|Q|=q,P\cap Q=\emptyset}} a_P^Q$$

denote the total number of bits present exactly in q nodes and demanded exactly by p (other) nodes. Then,

$$L^* \geq \sum_{p=1}^{K-1} \sum_{q=1}^{K-p} \frac{p}{p + q - 1} n(p, q). \quad (2)$$

Remark 1. *In [13], the authors presented an essentially identical bound (Lemma 1, [13]) as Corollary 1 in the setting of coded distributed computing. The proof given in [13] for this lemma also generalizes the arguments presented in [3], as does this work. Our present work considers a general data exchange problem and derives the lower bound in Theorem 1 for the communication load in such a setting. We had derived this lower bound independently in the conference version of this paper [14], and only recently came to know about the bound in [13]. In subsequent sections, we show how to use this bound to recover converses for various multi-terminal communication problems considered in the literature in recent years, and also obtain new converses for some settings. We also discuss, in Section 5, the looseness of Theorem 1 by considering a centralized version of the data exchange problem and comparing our bound with the generalized independence number bound in index coding. In Section 6, we discuss the application of our data exchange bound to more generalized index coding settings. These are the novel features of our present work, compared to the bound in Lemma 1 of [13].*

1.2. A Generic Outline of the Converse Proofs Presented in This Paper

In this work, we derive converse bounds for various settings in coded caching, coded distributed computing, and coded data shuffling using the bound in Theorem 1. Some of these converse bounds are already available in the literature, while others are novel. Each setting enjoins some constraints on the size of the demands and the size of the pre-stored content at each node. The bound in Theorem 1 applies for the setting in which the nodes have some predetermined local storage and some specific demanded bits. However, the settings of coded caching, coded distributed computing, and coded data shuffling permit the design of the initial storage so that the communication load is minimized. Further, the optimal communication load as defined in the literature for some of these settings involves maximization over all possible demand configurations, keeping only the size of the demands fixed. Keeping with these specifics, our bound in Theorem 1 must be tuned for each setting to obtain the respective converse, as captured by the three following steps which describe the generic structure behind our converse proofs.

1. **Applying Theorem 1** to the present setting, we obtain a lower bound expression on the communication load, assuming an arbitrary choice of demands across the nodes and some arbitrary but fixed storage across the nodes.
2. **"Symmetrization" step:** In this step, the lower bound expression obtained in the previous step is averaged over some carefully chosen configurations of demanded bits at the nodes. This step helps to remove the dependency of the lower bound on the specific choice of demands.
3. **Refine the averaged bound** by imposing the constraints on the size of the initial storage at the nodes, and using convexity of terms inside the averaged bound to obtain the final expression of the bound. This step helps to remove the dependency of the converse on the specific initial storage configuration at the nodes.

These three steps enable us to give simpler proofs to those in the literature for known converses, and also obtain novel converses for some variants of the same problems. Further, it also illustrates the generic nature of the data exchange bound of Theorem 1. In the converse proofs that are to follow in this paper, we will highlight these steps at the appropriate junctures.

2. Coded Caching

In this section, we apply Theorem 1 to recover the lower bound obtained in [15] for the problem of coded caching introduced in [1]. Further, using Theorem 1, we prove in Section 2.1 a new converse for a generic coded caching problem under multiple cache size settings. This provides new converses for some existing settings in literature, and also tightens bounds in some others. In Section 2.2, we recover a converse for coded caching with multiple file requests. In Section 2.3, we recover the converse for coded caching with decentralized cache placement.

We now describe the main setting of this section. In the coded caching system introduced in [1], there is one server connected via a noiseless broadcast channel to K clients indexed as $[K]$. The server possesses N files, each of size F bits, where the files are indexed as $W_i : i \in [N]$. Each client contains local storage, or a *cache*, of size MF bits, for some $M \leq N$. We call this a (K, M, N, F) coded caching system. Figure 1 illustrates this system model.

The coded caching system operates in two phases: in the *caching phase* which occurs during the low-traffic periods, the caches of the clients are populated by the server with some (uncoded) bits of the file library. This is known as *uncoded prefetching*. In this phase, the demands of the clients are not known. We denote the caching function for node k as ζ_k, and thus the cache content at client k at the end of the caching phase is denoted as $Z_k \triangleq \zeta_k(\{W_i : i \in [N]\})$.

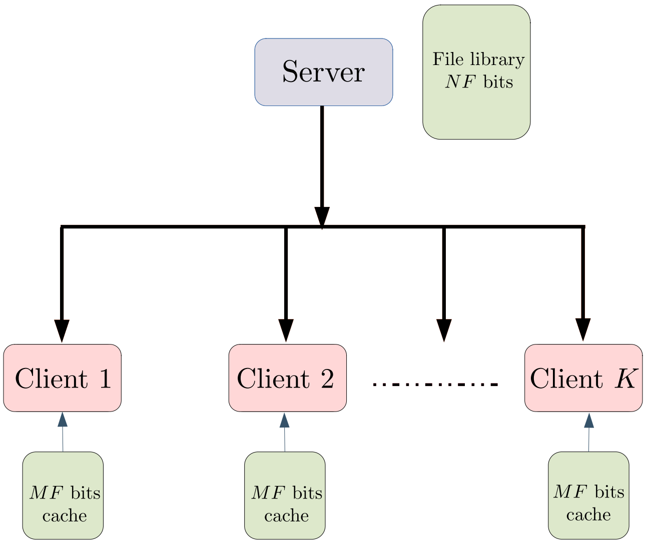

Figure 1. A single server is connected to K clients via a broadcast channel. Each user has a cache capable of storing MF of the NF bits in the file-library available at the server.

In the *delivery phase* which occurs during the high-traffic periods, each client demands one file from the server, and the server makes transmissions via the broadcast channel to satisfy the client demands. Let the demanded file at client k be W_{d_k}, where $d_k \in [N]$. The server uses an encoding function ϕ to obtain coded transmissions $X = \phi(\{W_{d_k} : k \in [K]\})$ such that each client $k \in [K]$ can employ a decoding function ψ_k to decode its demanded file using the coded transmissions and its cache content, i.e., $\psi_k(X, Z_k) = W_{d_k}$.

The communication load $L_c(\{\zeta_k : k \in [K]\}, \phi, \{\psi_k : k \in [K]\})$ of the above coded caching scheme is the number of bits transmitted in the delivery phase (i.e., the length of X) in the worst case (where "worst case" denotes maximization across all possible demands). The optimal communication load denoted by L_c^*, is then defined as

$$L_c^* \triangleq \min_{\{\zeta_k : k\in[K]\}, \phi, \{\psi_k : k \in [K]\}} L_c(\{\zeta_k : k \in [K]\}, \phi, \{\psi_k : k \in [K]\}). \tag{3}$$

For this system model, when $\frac{MK}{N} \in \mathbb{Z}$, the work in [1] proposed a caching and delivery scheme which achieves a communication load (normalized by the size of the file F) given by $K(1 - \frac{M}{N}) \min\left\{\frac{1}{1+\frac{MK}{N}}, \frac{N}{K}\right\}$. In [15], it was shown that, for any coded caching scheme with uncoded cache placement, the optimal communication load is lower bounded by $L_c^* \geq \frac{K(1-\frac{M}{N})F}{1+MK/N}$. Therefore, it was shown that, when $K \leq N$ and $\frac{MK}{N} \in \mathbb{Z}$, the scheme in [1] is optimal.

In the present section, we give another proof of the lower bound for coded caching derived in [15]. We later discuss the case of arbitrary K, N in Remark 2.

We now proceed with restating the lower bound from [15]. Note that these converses are typically normalized by the file size in literature, however we recall them in their non-normalized form, in order to relate them with our data exchange problem setting.

Theorem 2 ([15]). *Consider a (K, M, N, F) coded caching system with $K \leq N$. The optimal communication load L_c^* in the delivery phase satisfies*

$$L_c^* \geq \frac{K(1 - M/N)}{1 + MK/N} F.$$

Proof based on Theorem 1. We assume that the caching scheme and delivery scheme of the coded caching scheme are designed such that the communication load L_c is exactly equal to the optimal load L_c^*. Let the K client demands in the delivery phase be represented by a demand vector $\boldsymbol{d} = (d_1, \ldots, d_K)$, where $d_k \in [N]$ denotes the index of the demanded file of the client k. We are interested in the worst case demands scenario; this means we can assume that all the demanded files are distinct, i.e., $d_k \neq d_{k'}$ for all $k \neq k'$ to bound L_c^* from below, without loss of generality.

We observe that a (K, M, N, F) coded caching problem during the delivery phase satisfies Definition 1 of a data exchange problem on $K+1$ nodes indexed as $\{0, 1, \ldots, K\}$, where we give the index 0 to the server node and include this in the data exchange system. Before proceeding, we remark that the below proof gives a lower bound where all $K+1$ nodes in the system *may* transmit, whereas in the coded caching system of [1] *only* the server *can* transmit. Thus, any lower bound that we obtain in this proof applies to the setting in [1] also.

Clearly in the equivalent data exchange problem, the node 0 (the server) does not demand anything, but has a copy of all the bits in the entire system. With these observations, we have by definition of a_P^Q in (1)

$$a_P^Q = 0, \quad \text{if } 0 \notin Q \text{ or if } P \notin \binom{[K]}{1}, \tag{4}$$

where the quantities a_P^Q clearly depend on the demand vector $\boldsymbol{d}$.

We thus use a new set of variables: for each $k \in [K]$, $Q \subset [K]$, and given demands $\boldsymbol{d} = (d_1, \ldots, d_K)$, let $c_k^Q(\boldsymbol{d})$ denote the number of bits demanded by receiver node k that are available only at the nodes $Q \cup \{0\}$, i.e.,

$$c_k^Q(\boldsymbol{d}) \triangleq a_{\{k\}}^{Q \cup 0}. \tag{5}$$

Using these definitions, we proceed following the three steps given in Section 1.2.

Applying Theorem 1: By Theorem 1, we have the following lower bound for demand vector $\boldsymbol{d}$

$$\begin{aligned} L_c^* = L_c &\geq \sum_{P \subset [K] \cup \{0\}} \sum_{Q' \subset [K] \cup \{0\} \backslash P} \frac{|P|}{|P| + |Q'| - 1} a_P^{Q'} \\ &= \sum_{k=1}^{K} \sum_{Q \subset [K] \backslash k} \frac{1}{|Q| + 1} c_k^Q(\boldsymbol{d}), \end{aligned} \tag{6}$$

where (6) is obtained from (4) and (5).

"Symmetrizing" (6) over carefully chosen demand vectors: We now consider the averaging of bounds of type (6) over a chosen collection of N demand vectors, given by

$$\mathcal{D} \triangleq \left\{ \left(j \oplus_N 0, j \oplus_N 1, \ldots, j \oplus_N (K-1) \right) : j = 0, \ldots, N-1 \right\} \tag{7}$$

where $j \oplus_N i \triangleq ((j+i) \mod N) + 1$.

That is, $\mathcal{D}$ contains the demand vectors consisting of consecutive K files, starting with each of the N files as the demand of the first client.

Averaging (6) through the set of N demand vectors in $\mathcal{D}$, the lower bound we obtain is

$$L_c^* \geq \frac{1}{N} \sum_{\boldsymbol{d} \in \mathcal{D}} \sum_{k=1}^{K} \sum_{Q \subset [K] \backslash k} \frac{1}{|Q| + 1} c_k^Q(\boldsymbol{d}). \tag{8}$$

Let b_n^Q denote the number of bits of file n stored only in $Q \cup \{0\}$. Then, in the above sum, $b_n^Q = c_k^Q(\boldsymbol{d})$ if and only if $d_k = n$. This happens precisely once in the collection of N demand vectors in $\mathcal{D}$. Thus, we have

$$\begin{aligned} L_c^* &\geq \frac{1}{N}\sum_{k=1}^{K}\sum_{Q\subset[K]\setminus k}\sum_{\boldsymbol{d}\in\mathcal{D}}\frac{1}{|Q|+1}c_k^Q(\boldsymbol{d}) \\ &= \frac{1}{N}\sum_{k=1}^{K}\sum_{Q\subset[K]\setminus k}\sum_{n=1}^{N}\frac{1}{|Q|+1}b_n^Q \quad (9) \\ &= F\sum_{Q\subset[K]}\sum_{n=1}^{N}\left(\frac{K-|Q|}{|Q|+1}\right)\frac{b_n^Q}{NF} \quad (10) \end{aligned}$$

where (10) follows as for a fixed n and Q, $k \in [K]\setminus Q$ in (9), and by multiplying and dividing by F.

Refining the bound (10) by using the constraints of the setting: Now, by definition, $\sum_n \sum_{Q\subset[K]} b_n^Q = NF$, and thus $b_n^Q/NF : n \in [N], Q \subset [K]$, denotes a probability mass function. Furthermore, $\sum_{Q\subset[K]} |Q| b_n^Q \leq KMF$. As $(K-x)/(1+x)$ is a convex decreasing function for $x \geq 0$, using Jensen's inequality, we have $L_c^* \geq (K-x)/(1+x)$, where

$$x = \sum_n \sum_{Q\subset[K]} |Q|\frac{b_n^Q}{NF} \leq \frac{KMF}{NF} = \frac{KM}{N}.$$

Thus, we get $L_c^* \geq \frac{K(1-M/N)}{1+MK/N}F$, which completes the proof. □

Remark 2. *In the previous part of this section, we have shown the converse for the worst case communication load L_c^* for coded caching in the regime of $K \leq N$. We now consider a general coded caching setup with arbitrary K, N values and cache size M. Consider a positive integer $N_u \leq \min\{N, K\}$. For a fixed caching scheme denoted by $\zeta = \{\zeta_k : k \in [K]\}$, let the minimum communication load for satisfying the clients, maximized across all possible demand vectors with exactly N_u distinct files in each of the demand vectors, be denoted as $L_c^*(N_u, \zeta)$.*

In the work [16], it was shown that for $t \triangleq \frac{MK}{N}$,

$$L_c^*(N_u, \zeta) \geq g_{N_u}(t), \tag{11}$$

where $g_{N_u}(x)$ is defined as the lower convex envelope of the points

$$\mathcal{P}(N_u) = \left\{\left(x, \frac{\binom{K}{x+1} - \binom{K-N_u}{x+1}}{\binom{K}{x}}F\right) : x \in \{0, \ldots, K\}\right\}.$$

Note that $g_{N_u}(t)$ is independent of ζ. For this general setting, the optimal worst case load L_c^*, as defined in (3), satisfies

$$L_c^* = \min_{\zeta} L_c^*(\min\{N, K\}, \zeta).$$

Thus, from (11), we get

$$L_c^* \geq g_{\min\{N,K\}}(t), \tag{12}$$

which is the converse bound on the worst case communication load proved in [16] for this general scenario. In Appendix B, we use our data exchange bound in Theorem 1 to recover (11), which therefore shows (12).

2.1. Server-Based and Server-Free Coded Caching with Heterogeneous Cache Sizes at Clients

So far we have discussed the coded caching scenario where there is a central server containing the entire file library and the client cache sizes are homogeneous, i.e., the same at all clients. We now describe a generalization of the result in Theorem 2 to the case of systems in which the clients have heterogeneous cache sizes, with either a centralized server present or absent. The proof of this is easily obtained from our data exchange bound in Theorem 1. To the best of our knowledge, a converse for this general setting is not known in the literature. Using this converse, we can derive new converses and tighten existing converses for various special cases of this setting, which include widely studied coded caching settings, such as device-to-device coded caching [17].

Consider a coded caching system with N files (each of size F) with K client nodes denoted by a set $\mathcal{K}_T$. We shall indicate by the value γ the presence ($\gamma = 1$) or absence ($\gamma = 0$) of a centralized server in the system containing the file library. For the purpose of utilizing our data exchange bound, we assume that all the nodes in the system are capable of transmissions; thereby, any converse for this scenario is also valid for the usual coded caching scenario in which only the server (if it is present) does transmissions in the delivery phase. The set of clients $\mathcal{K}_T$ is partitioned into subsets $\mathcal{K}_{T_i} : i = 1, \dots, t$ where the nodes in subset $\mathcal{K}_{T_i}$ can store a fraction γ_{T_i} of the file library. Let $|\mathcal{K}_{T_i}| = K_{T_i}$. We now give our converse for this setting. The caching and the delivery scheme, as well as the optimal communication load L_c^*, are defined as in the case of coded caching with homogeneous cache sizes.

Proposition 1. *For the above heterogeneous cache sizes setting, assuming $K \leq N$, the optimal communication load L_c^* for uncoded cache placement is lower bounded as follows.*

$$L_c^* \geq \left(\frac{K - \sum_{i=1}^{t} K_{T_i}\gamma_{T_i}}{\gamma + \sum_{i=1}^{t} K_{T_i}\gamma_{T_i}} \right) F. \tag{13}$$

Before giving the proof of Proposition 1, we give the following remarks regarding the generality of Proposition 1, the new results which arise by applying Proposition 1 and various results from existing literature that are subsumed or improved by it.

- *Heterogeneous Cache Sizes:* There exists a number of works discussing distinct or heterogenous client cache sizes, for instance, in [18,19]. However, closed form expressions for the lower bound on the load seem to be lacking for such scenarios, to the best of our knowledge. Proposition 1 gives a lower bound for all such settings.
- *Device-to-Device Coded Caching:* Suppose there is no designated server in a coded caching setup, but the client nodes themselves are responsible for exchanging the information to satisfy their demands. This corresponds to the case of Device-to-Device (D2D) coded caching, first explored in [17]. In [17], an achievable scheme was presented for the case when each (client) node has equal cache fraction $\frac{M}{N}$, and this scheme achieves a communication load of $(\frac{N}{M} - 1)F$ bits. In the work [20], it was shown that this communication load is optimal (for the regime of $K \leq N$) over all possible "one shot" schemes (where "one shot" refers to those schemes in which each demanded bit is decoded using the transmission only from one server), and further it was shown that the load is within a multiplicative factor of 2 of the optimal communication load under the constraint for uncoded cache placement. We remark that the D2D setting of [17] corresponds to the special case of our current setting, with $\gamma = 0,\ t = 1,\ K_{T_1} = K,$ and $\gamma_{T_1} = M/N$. By this correspondence, by applying Proposition 1, we see that the load in this case is lower bounded as $\left(\frac{N}{M} - 1\right)F$, thus showing that the achievable scheme in [17] is exactly optimal under uncoded cache placement. The D2D scenario with heterogeneous cache sizes was explored in [21], in which the optimal communication load was characterized as the solution of an optimization problem. However, no closed form expression of the load for such a

scenario is mentioned. Clearly, our Proposition 1 gives such a lower bound, when we fix $\gamma = 0$, for any number of levels t of the client-side cache sizes.

Further, the result for coded caching with a server and equal cache sizes at receivers, as in Theorem 2, is clearly obtained as a special case of Proposition 1 with $\gamma = 1,\ t = 1,\ K_{T_1} = K$ and $\gamma_{T_1} = \frac{M}{N}$.

We now proceed to prove Proposition 1. The proof is similar to that of Theorem 2.

Proof of Proposition 1. As in the proof of Theorem 2, we will denote the server node as the node 0 and assume a caching and delivery scheme which achieves the optimal load L_c^* for worst case client demands.

Applying Theorem 1, for our setting, we have

$$L_c^* \geq \begin{cases} \sum_{P \subset \mathcal{K}_T} \sum_{Q \subset \mathcal{K}_T \setminus P} \frac{|P|}{|P|+|Q|-1} a_P^Q & \text{if } \gamma = 0, \\ \sum_{P \subset \mathcal{K}_T} \sum_{Q' \subset \mathcal{K}_T \cup 0 \setminus P} \frac{|P|}{|P|+|Q'|-1} a_P^{Q'} & \text{if } \gamma = 1. \end{cases}$$

Note that if $\gamma = 1$ (i.e., the server is present), then $a_P^{Q'} = 0$ whenever $0 \notin Q'$.

For a specific demand vector $\boldsymbol{d} = (d_1, \ldots, d_K)$ consisting of distinct demands and for some $Q \subset \mathcal{K}_T$, we define the quantity $c_k^Q(\boldsymbol{d})$ as follows.

$$c_k^Q(\boldsymbol{d}) = \begin{cases} a_{\{k\}}^Q = \text{Number of bits demanded by } k & \text{if } \gamma = 0, \\ \qquad \text{available exclusively in } Q & \\ a_{\{k\}}^{Q \cup 0} = \text{Number of bits demanded by } k & \text{if } \gamma = 1. \\ \qquad \text{available exclusively in } Q \cup 0 & \end{cases}$$

Symmetrization over appropriately chosen demand vectors: Choosing the same special set of demand vectors $\mathcal{D}$ as in (7) and averaging the above lower bound over the demand vectors in $\mathcal{D}$ similar to the proof of Theorem 2, we obtain a bound similar to (8):

$$L_c^* \geq \begin{cases} \frac{1}{N} \sum_{\boldsymbol{d} \in \mathcal{D}} \sum_{k \in \mathcal{K}_T} \sum_{Q \subset \mathcal{K}_T \setminus k} \frac{c_k^Q(\boldsymbol{d})}{|Q|} & \text{if } \ \gamma = 0, \\ \frac{1}{N} \sum_{\boldsymbol{d} \in \mathcal{D}} \sum_{k \in \mathcal{K}_T} \sum_{Q \subset \mathcal{K}_T \setminus k} \frac{c_k^Q(\boldsymbol{d})}{|Q \cup 0|} & \text{if } \ \gamma = 1. \end{cases} \tag{14}$$

Combining the two expressions in (14), we can write a single equation which holds for $\gamma \in \{0, 1\}$,

$$L_c^* \geq \frac{1}{N} \sum_{\boldsymbol{d} \in \mathcal{D}} \sum_{k \in \mathcal{K}_T} \sum_{Q \subset \mathcal{K}_T \setminus k} \frac{c_k^Q(\boldsymbol{d})}{\gamma + |Q|}. \tag{15}$$

We now define the term b_n^Q as follows.

$$b_n^Q = \begin{cases} \text{Number of bits of file } n \text{ available exclusively in } Q & \text{if } \gamma = 0, \\ \text{Number of bits of file } n \text{ available exclusively in } Q \cup 0 & \text{if } \gamma = 1. \end{cases} \tag{16}$$

Using the above definition of b_n^Q and observing that each demand vector in $\mathcal{D}$ has distinct components, Equation (15) can be written as

$$L_c^* \geq \frac{1}{N} \sum_{k\in\mathcal{K}_T} \sum_{Q\subset\mathcal{K}_T\setminus k} \sum_{\boldsymbol{d}\in\mathcal{D}} \frac{c_k^Q(\boldsymbol{d})}{\gamma+|Q|} \tag{17}$$

$$= \frac{1}{N} \sum_{k\in\mathcal{K}_T} \sum_{Q\subset\mathcal{K}_T\setminus k} \sum_{n=1}^{N} \frac{b_n^Q}{\gamma+|Q|} \tag{18}$$

$$= \frac{1}{N} \sum_{Q\subset\mathcal{K}_T} \sum_{n=1}^{N} \frac{(K-|Q|)b_n^Q}{\gamma+|Q|}. \tag{19}$$

Refining the bound in (19) using setting constraints and convexity: By the definition of b_n^Q in (16), we have $\sum_n \sum_{Q\subset\mathcal{K}_T} b_n^Q = NF$. Further, $\sum_{Q\subset\mathcal{K}_T} |Q| b_n^Q \leq \sum_{i=1}^{t} K_{T_i}\gamma_{T_i}NF$. Furthermore, for $\gamma \geq 0$, the function $\frac{K-x}{\gamma+x}$ is a convex decreasing function in x for $x > 0$. Thus, using Jensen's inequality, we have $L_c^* \geq \frac{K-x}{\gamma+x}$, where

$$x = \sum_n \sum_{Q\subset\mathcal{K}_T} |Q| \frac{b_n^Q}{NF} \leq \frac{\sum_{i=1}^{t} K_{T_i}\gamma_{T_i}NF}{NF} = \sum_{i=1}^{t} K_{T_i}\gamma_{T_i}.$$

This completes the proof. □

Remark 3. *Proposition 1 holds when $N \geq K$. This scenario is the most studied case in the literature and is practically more relevant than the case $K > N$. We now provide lower bounds for the heterogeneous cache sizes setting for general values of K, N, which includes the case $K > N$. As before, we consider two cases: $\gamma = 1$ indicates the presence of a centralized server in the system and $\gamma = 0$ indicates its absence.*

Case 1, $\gamma = 1$: *For the case where a centralized server is present, i.e., $\gamma = 1$, we have*

$$L_c^* \geq g_{\min\{N,K\}}\left(\sum_{i=1}^{t} K_{T_i}\gamma_{T_i}\right), \tag{20}$$

where the function $g_{\min\{N,K\}}$ is defined in Remark 2. The derivation of this lower bound follows the steps in Appendix B until (A21), where we choose $N_u = \min\{N,K\}$. Without loss of generality, we assume that all caches are fully populated with uncoded bits from the library, thus the total memory occupied by the cached bits $\sum_{Q\subset[K]} |Q|a^Q$ is equal to the sum of all the cache memory available in the system $\sum_{i=1}^{t} K_{T_i}\gamma_{T_i}NF$. Applying Jensen's inequality on (A21) and using the fact $\sum_{Q\subset[K]} |Q|\frac{a^Q}{NF} = \sum_{i=1}^{t} K_{T_i}\gamma_{T_i}$, we immediately arrive at the lower bound (20).

Case 2, $\gamma = 0$: *In this case, the optimal worst-case communication load can be lower bounded as follows:*

$$L_c^* \geq \frac{\min\{N,K\}}{K}\left(\frac{K-\sum_{i=1}^{t} K_{T_i}\gamma_{T_i}}{\sum_{i=1}^{t} K_{T_i}\gamma_{T_i}}\right)F. \tag{21}$$

The proof of this lower bound follows similar approach as Appendix B and is outlined in Appendix C. Note that when $N \geq K$, both (20) and (21) become identical to the inequality in Proposition 1.

2.2. Coded Caching with Multiple File Requests

In [22], coded caching with multiple file requests was considered, in which each client requests any Δ files out of the N files in the delivery phase. It was shown in [22] (Section V.A) that if the $\Delta K \leq N$, then the optimal worst case communication load can be lower bounded as

$$L_c^* \geq \frac{K\Delta(1-M/N)}{1+MK/N}F. \tag{22}$$

The work in [22] also gives an achievable scheme based on the scheme in [1] which meets the above bound. The same lower bound can be derived using Theorem 1 also, by following a similar procedure as that of the proof of Theorem 2.

Applying Theorem 1, we give the proof in brief. The demand vector assumed in proof of Theorem 1 becomes a $K\Delta$-length vector in this case, consisting of K subvectors, each of length Δ, capturing Δ distinct demands for each client. The proof proceeds as is until (6).

Symmetrization: The set $\mathcal{D}$ in (7) now contains the $K\Delta$-length vectors of consecutive file indices, cyclically constructed, starting from $(1, \ldots, K\Delta)$, i.e.,

$$\mathcal{D} \triangleq \Big\{ \Big(j \oplus_N 0, j \oplus_N 1, \ldots, j \oplus_N (K\Delta - 1) \Big) : j = 0, \ldots, N-1 \Big\}. \tag{23}$$

Thus, if the demand vector considered is $\boldsymbol{d}(j) \triangleq \Big(j \oplus_N 0, j \oplus_N 1, \ldots, j \oplus_N (K\Delta - 1) \Big) \in \mathcal{D}$, then the indices of the demanded files at client $k \in [K]$, denoted by $\boldsymbol{d}_k(j)$, is given by

$$\boldsymbol{d}_k(j) \triangleq \{ j \oplus_N (k-1)\Delta, j \oplus_N (k-1)\Delta + 1, \ldots, j \oplus_N (k\Delta - 1) \}.$$

The averaged lower bound expression similar to (8) is then obtained as

$$L_c^* \geq \frac{1}{N} \sum_{j=0}^{N-1} \sum_{k=1}^{K} \sum_{Q \subset [K] \setminus k} \frac{1}{|Q|+1} c_k^Q(\boldsymbol{d}(j)). \tag{24}$$

In this expression, we have $c_k^Q(\boldsymbol{d}(j))$ which now indicates the number of bits of Δ distinct and consecutive files indexed by $\boldsymbol{d}_k(j)$ and available exclusively at the nodes in $Q \cup 0$ (0 denoting the server).

Observation: $c_k^Q(\boldsymbol{d}(j)) = \sum_{n' \in \boldsymbol{d}_k(j)} b_{n'}^Q$ where $b_{n'}^Q$ denotes the number of bits of file n' available exclusively in the nodes $Q \cup 0$, as in the proof of Theorem 2.

Now, $n' \in \boldsymbol{d}_k(j)$ if and only if the file n' is demanded by client k. By definition of $\mathcal{D}$, the event $n' \in \boldsymbol{d}_k(j)$ happens for precisely Δ values of index j. From (24), applying the above observation, we have the following.

$$\begin{aligned} L_c^* &\geq \frac{1}{N} \sum_{k=1}^{K} \sum_{Q \subset [K] \setminus k} \sum_{j=0}^{N-1} \frac{1}{|Q|+1} c_k^Q(\boldsymbol{d}(j)) \\ &= \frac{1}{N} \sum_{k=1}^{K} \sum_{Q \subset [K] \setminus k} \frac{1}{|Q|+1} \left(\sum_{j=0}^{N-1} \sum_{n' \in \boldsymbol{d}_k(j)} b_{n'}^Q \right) \\ &= \frac{1}{N} \sum_{k=1}^{K} \sum_{Q \subset [K] \setminus k} \sum_{n'=1}^{N} \frac{\Delta}{|Q|+1} b_{n'}^Q. \end{aligned} \tag{25}$$

Refining the bound in (25) using the setting constraints: We use the constraints of the setting and the convexity of the resultant expression to refine (25). This refinement essentially follows similar subsequent steps as in the proof of Theorem 2 following (9), and leads finally to (22).

Remark 4. *The work in [23] considers a coded caching setup in which Λ caches ($\Lambda \leq K$) are shared between the K clients. The special case when Λ divides K and each cache is serving exactly $\frac{K}{\Lambda}$ clients is equivalent to the scenario of the multiple file requests in [22] with Λ clients, each demanding $\frac{K}{\Lambda}$ files. The above proof then recovers the converse for this setting, which is obtained in [23] (Section III.A in [23]).*

2.3. Coded Caching with Decentralized Caching

Theorem 2 and the subsequent results discussed above hold for the *centralized caching* framework, in which the caching phase is designed carefully in a predetermined fashion. In [24], the idea of decentralized placement was introduced, in which the caching phase is not coordinated centrally (this was called “decentralized coded caching” in [24]). In this scenario, each client, independently of others, caches a fraction $\gamma = \frac{M}{N}$ of the bits in each of the N files in the file library, chosen uniformly at random. For this scenario, the server (which has the file library) is responsible for the delivery phase. The optimal communication load L_c^* is defined as the minimum worst case communication load over all possible delivery schemes for a given caching configuration, randomly constructed as given above. For the case of $K \leq N$, the authors of [24] show a scheme which achieves the worst case communication load $L_c = KF(1-M/N)\frac{(1-(1-M/N)^K)}{MK/N}$. This was shown to be optimal for large F in [16] and also in [25] via a connection to index coding. In the following, we show that the same optimality follows easily via our Theorem 1.

Assume that we have distinct demands at the K clients, as in the proof of Theorem 1, given by the demand vector $\boldsymbol{d}$. We first note that by the law of large numbers, as F increases, for the decentralized cache placement, for any $k \in [K], Q \subset [K] \setminus k$, we have

$$c_k^Q(\boldsymbol{d}) = F\left(\frac{M}{N}\right)^{|Q|}\left(1-\frac{M}{N}\right)^{K-|Q|},$$

with probability close to 1, where $c_k^Q(\boldsymbol{d})$ is as defined in (5). This observation enables us to avoid the steps 2 and 3 mentioned in Section 1.2, as the value of $c_k^Q(\boldsymbol{d})$ is independent of the specific random cache placement or the demands chosen (as long as they are distinct). Using this in (6), we get

$$\begin{aligned}
L_c^* &\geq \sum_{k=1}^{K}\sum_{Q\subset[K]\setminus k}\frac{1}{|Q|+1}c_k^Q(\boldsymbol{d}) \\
&\geq \sum_{k=1}^{K}\sum_{Q\subset[K]\setminus k}\frac{F}{|Q|+1}\left(\frac{M}{N}\right)^{|Q|}\left(1-\frac{M}{N}\right)^{K-|Q|} \\
&= F\sum_{Q\subset[K]}\left(\frac{K-|Q|}{|Q|+1}\right)\left(\frac{M}{N}\right)^{|Q|}\left(1-\frac{M}{N}\right)^{K-|Q|} \\
&= F\sum_{i=0}^{K}\binom{K}{i}\left(\frac{K-i}{i+1}\right)\left(\frac{M}{N}\right)^{i}\left(1-\frac{M}{N}\right)^{K-i} \\
&= F\sum_{i=0}^{K}\binom{K}{i+1}\left(\frac{M}{N}\right)^{i}\left(1-\frac{M}{N}\right)^{K-i} \\
&= F\sum_{j=1}^{K}\binom{K}{j}\left(\frac{M}{N}\right)^{j-1}\left(1-\frac{M}{N}\right)^{K-j+1} \\
&= F\frac{N}{M}\left(1-\frac{M}{N}\right)\sum_{j=1}^{K}\binom{K}{j}\left(\frac{M}{N}\right)^{j}\left(1-\frac{M}{N}\right)^{K-j} \\
&= \frac{NF}{M}\left(1-\frac{M}{N}\right)\left(1-\left(1-\frac{M}{N}\right)^{K}\right),
\end{aligned}$$

where the last step follows as $\sum_{j=0}^{K}\binom{K}{j}\left(\frac{M}{N}\right)^{j}\left(1-\frac{M}{N}\right)^{K-j} = \left(\frac{M}{N}+\left(1-\frac{M}{N}\right)\right)^{K} = 1$. Thus, we have given an alternate proof of the optimality of the decentralized scheme in [24].

3. Decentralized Coded Data Shuffling

In distributed machine learning systems consisting of a master and multiple worker nodes, data are distributed to the workers by the master in order to perform training of the machine learning model in a distributed manner. In general, this training process takes multiple iterations, with the workers doing some processing (like computing gradients) on their respective training data subsets. In order to ensure that the training data subset at each node are sufficiently representative of the data, and to improve the statistical performance of machine learning algorithms, shuffling of the training data between the worker nodes is implemented after every training iteration. This is known as *data shuffling*.

A coding theoretic approach to data shuffling, which involves the master communicating coded data to the workers was presented in [4]. The setting in [4] was *centralized*, which meant that there is a master node communicating to the servers to perform the data shuffling.

The work in [5] considered the data shuffling problem in which there is no master node, but the worker nodes exchange the training data among themselves, without involving the master node, to create a new desired partition in the next iteration. This was termed as *decentralized data shuffling* in [5]. Note that these notions of "centralized" and "decentralized" in the data shuffling problem are different from those in the coded caching [24], in which these terms were used to define the deterministic and random design of the caching phase, respectively. In this section, we look at the work in [5] and give a new simpler proof of the lower bound on the communication load for decentralized data shuffling.

We first review the setting in [5]. Consider K workers in the system, where each worker node is required to process q data units at any given time. The total dataset $F_1 \cup \cdots \cup F_N$ consists of $N = Kq$ data units $F_1, \ldots, F_N$, with a size of B bits per data unit. The collection of data units to be processed by worker node k at time t is denoted as $A_{k,t}$. The collection of data units $A_{1,t}, \ldots, A_{K,t}$ must form a partition of the dataset $F_1 \cup \cdots \cup F_N$ for every time instant t, i.e., for any time t and any choice of $k, k' \in [K]$ with $k \neq k'$ we have

$$A_{k,t} \subset F_1 \cup \cdots \cup F_N, |A_{k,t}| = qB \text{ and } A_{k,t} \cap A_{k',t} = \emptyset.$$

Each node k has a local cache of size MB bits (such that $q \leq M \leq Kq$) that can hold M data units. Out of these M units q units are the current "active" data $A_{k,t}$ at any time step which are required to be processed by the node k. The contents of the cache of node k at time t is denoted as $Z_{k,t}$. Therefore, for each choice of $k \in [K]$ and any time t, we have

$$|Z_{k,t}| = MB \text{ and } A_{k,t} \subset Z_{k,t}.$$

At each time instance t, a new partition $\{A_{k,t} : k \in [K]\}$ is to be made active at the nodes $[K]$, where this new partition is made known to the workers only at time step t. Note that the contents of the nodes at time $t-1$ are $Z_{1,t-1}, \ldots, Z_{K,t-1}$, and the active partition at time $t-1$ is $A_{1,t-1}, \ldots, A_{K,t-1}$. The worker nodes communicate with each other over a common broadcast link, as shown in Figure 2, to achieve the new partition. The decentralized data shuffling problem is to find a delivery scheme (between workers) to shuffle the collection of active data units $\{A_{k,t-1} : k \in [K]\}$ to a new partition $\{A_{k,t} : k \in [K]\}$. Each worker k computes a function $\phi_k(Z_{k,t-1})$ of its cache contents and broadcasts it to the other workers. Using these transmissions and the locally available cache content $Z_{k,t-1}$, each node k is required to decode $A_{k,t}$. As in the case of coded caching, one seeks to reduce the worst-case communication load by designing the initial storage and coded transmissions carefully. The communication load of this data shuffling scheme, denoted by L_{ds}, is the sum of the number of bits broadcast by all the K nodes in the system, i.e., $L_{ds} = \sum_{k \in [K]} |\phi_k(Z_{k,t-1})|$. The optimal communication load of data shuffling L_{ds}^* (for the worst case data shuffle) is defined as

$$L_{ds}^* = \min \max L_{ds}$$

where the maximization is over all possible choices for $A_{k,t-1} : k \in [K]$ and $A_{k,t} : k \in [K]$, and the minimization is over all possible choices for the cache placement $\{Z_{k,t-1} : k \in [K]\}$ and the delivery scheme $\{\phi_k : k \in [K]\}$.

For the above setting, the following bound on the communication load L_d^* was shown in [5].

$$L_{ds}^* \geq \frac{Kq}{K-1} \cdot \frac{K - M/q}{M/q} B. \tag{26}$$

The above bound was shown to be optimal for some special cases of the parameters, and order-optimal otherwise.

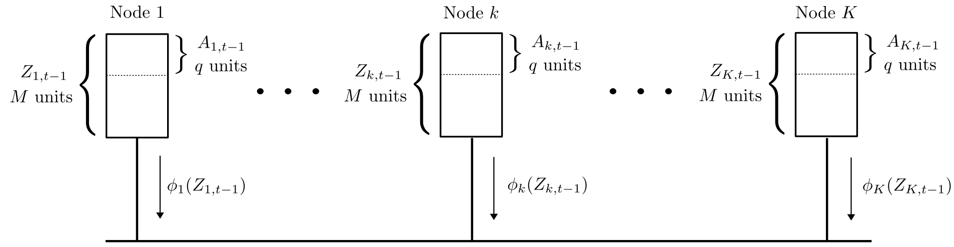

Figure 2. The decentralized data shuffling problem. The contents of the local cache of each node k at time $t-1$ is $Z_{k,t-1}$, out of which a subset $A_{k,t-1}$ is the active data currently processed by the node. The worker nodes must communicate via a broadcast link to shuffle the active data among each other and create a new partition $A_{1,t}, \ldots, A_{K,t}$ of the data units at the next time instance t.

Proof of the Decentralized Data Shuffling Converse

We now recover the bound (26) by a simple proof using our generic lower bound in Theorem 1. We assume that the cache placement and delivery scheme of the data shuffling scheme are designed such that the communication load of the data shuffling scheme is exactly equal to L_{ds}^*. We proceed as per the three steps in Section 1.2.

Applying Theorem 1: For $k \in [K]$ and $Q \subset [K]$, let $A_{k,t}^Q$ denote the subset of bits of $A_{k,t}$ available exactly at the nodes in Q and not anywhere else. Note that $|A_{k,t}^Q| = 0$ if $Q = \emptyset$, as each bit is necessarily present in at least one of the K nodes.

As per our bound in Theorem 1, we have

$$L_{ds}^* = L_{ds} \geq \sum_{k \in [K]} \sum_{Q \subset [K] \setminus k} \frac{|A_{k,t}^Q|}{|Q|}.$$

Symmetrization by averaging over appropriately chosen set of shuffles: Let the set of circular permutations of $(1, 2, \ldots, K)$, apart from the identity permutation, be denoted by Γ. There are $K-1$ of them clearly. We denote an arbitrary permutation in Γ by γ, and by γ_k we denote the kth coordinate of γ.

Now, consider the shuffle given by $\gamma \in \Gamma$, i.e., for each k, $A_{k,t} = A_{\gamma_k,t-1}$. For this shuffle, we have by the above equation that

$$L_{ds}^* \geq \sum_{k \in [K]} \sum_{Q \subset [K] \setminus k} \frac{|A_{\gamma_k,t-1}^Q|}{|Q|} \tag{27}$$

$$= \sum_{Q \subset [K]} \sum_{k \in [K] \setminus Q} \frac{|A_{\gamma_k,t-1}^Q|}{|Q|}. \tag{28}$$

Now, averaging (28) over all permutations in Γ, we get

$$L^*_{ds} \geq \frac{1}{K-1} \sum_{\gamma \in \Gamma} \sum_{Q \subset [K]} \sum_{k \in [K] \setminus Q} \frac{|A^Q_{\gamma_k, t-1}|}{|Q|}, \tag{29}$$

$$= \frac{1}{K-1} \sum_{Q \subset [K]} \sum_{k \in [K] \setminus Q} \sum_{\gamma \in \Gamma} \frac{|A^Q_{\gamma_k, t-1}|}{|Q|}. \tag{30}$$

As we go through all choices of $\gamma \in \Gamma$, we see that γ_k takes every value except k, i.e., γ_k assumes each value in $[K] \setminus k$ exactly once. Moreover, $A^Q_{k',t-1}$ is the collection of bits of $A_{k',t-1}$ present only in Q. However, the bits $A_{k',t-1}$ are already presented in k'. Hence, $|A^Q_{k',t-1}| = 0$ if $k' \notin Q$. Therefore, we have

$$L^*_{ds} \geq \frac{1}{K-1} \sum_{Q \subset [K]} \sum_{k \in [K] \setminus Q} \sum_{k' \in Q} \frac{|A^Q_{k',t-1}|}{|Q|}, \tag{31}$$

$$= \frac{1}{K-1} \sum_{Q \subset [K]} \sum_{k' \in Q} \frac{|A^Q_{k',t-1}|(K - |Q|)}{|Q|} \tag{32}$$

$$= \frac{1}{K-1} \sum_{k' \in [K]} \sum_{Q \subset [K] : k' \in Q} \frac{|A^Q_{k',t-1}|(K - |Q|)}{|Q|}. \tag{33}$$

Refining the bound using setting constraints and convexity: Now, we have the following observations as $A^Q_{k',t-1} : k' \in [K], \{Q' \subset [K] : k' \in Q\}$ form a partition of all the NB bits.

$$\sum_{Q \subset [K]} \sum_{k' \in Q} |A^Q_{k',t-1}| = NB = KqB$$
$$\sum_{Q \subset [K]} \sum_{k' \in Q} |A^Q_{k',t-1}||Q| \leq KMB.$$

Utilizing the above, and the fact that $\frac{K-|Q|}{|Q|}$ is a convex decreasing function in $|Q|$ (for $|Q| \geq 0$), we have

$$L^*_{ds} \geq \frac{KqB}{K-1} \cdot \frac{(K - \sum_{Q \subset [K]} \sum_{k' \in Q} \frac{|A^Q_{k',t-1}|}{NB} |Q|)}{\sum_{Q \subset [K]} \sum_{k' \in Q} \frac{|A^Q_{k',t-1}|}{NB} |Q|} \tag{34}$$

$$\geq \frac{KqB}{K-1} \cdot \frac{K - KM/N}{KM/N} \tag{35}$$

$$= \frac{KqB}{K-1} \cdot \frac{K - M/q}{M/q} \tag{36}$$

Thus, we have recovered (26).

Remark 5. *We have considered the decentralized version of the coded data shuffling problem in this subsection. The centralized version of the data shuffling problem was introduced in [4] and its information theoretic limits were studied elaborately in [6]. Our data exchange bound, when applied to the setting in [6], results in a looser converse result than that in [6]. The reasons for this is explored in Section 5 using the connection between our data exchange bound and the bound for index coding known in literature.*

4. Coded Distributed Computing

In a distributed computing setting, there are N files on which the distributed computing task has to be performed by K nodes. The job at hand is divided into three phases: Map, Shuffle, and Reduce. In the shuffle phase, the nodes that are assigned to perform the distributed computing task exchange data. In [3], the authors proposed coded communication during the shuffle phase to reduce the communication load. We recollect the setting and the main converse result from [3], which we recover using our data exchange bound.

A subset $\mathcal{M}_i$ of N files is assigned to ith node and the ith node computes the map functions on this subset in the map phase (see Figure 3). We assume that the total number of map functions computed at the K nodes is rN, where r is referred to as the *computation load*. In the reduce phase, a total of W reduce functions is to be computed across the K nodes corresponding to the N files. Each node is assigned the same number of functions. Obtaining the output of the reduce functions at all the nodes will complete the distributed computing task. In this work, as in [3], we consider two scenarios: in the first one, each reduce function is computed exactly at one node and in the second, each reduce function is computed at s nodes, where $s \geq 2$.

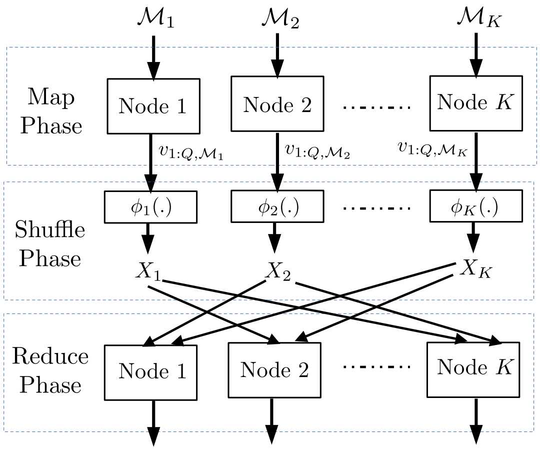

Figure 3. There are N files and a subset $\mathcal{M}_i$ of them are assigned to node i in map phase. The output of map phase at each node is $v_{1:Q,\mathcal{M}_i}$. Each node computes $X_i = \phi_i(v_{1:Q,\mathcal{M}_i})$ which it broadcasts to other nodes. The nodes compute the reduce outputs based on their own map outputs and the broadcasts which they receive.

Each map function output (also referred to as intermediate output) corresponds to a particular file and a particular reduce function. For each file and each reduce function, an intermediate output of T bits is obtained. To compute an assigned reduce function, each node requires the intermediate outputs of *all* the files corresponding to the assigned reduce function. This means each node is missing the intermediate outputs (corresponding to the assigned reduce functions) of those files that are not assigned to it in the map phase.

The intermediate outputs of each file assigned to node i corresponding to all the reduce functions are available at node i at the end of the map phase and denoted by $v_{1:Q,\mathcal{M}_i}$. These intermediate outputs at the end of the map phase are encoded as follows: $X_i = \phi_i(v_{1:Q,\mathcal{M}_i})$ and broadcasted to the remaining nodes, in the shuffle phase (in order to deliver the missing intermediate outputs at the nodes). Let L^*_{dc} be the total number of bits broadcasted by the K nodes in the shuffle phase, minimized over all possible map function assignments, reduce function assignments, and shuffling schemes, with a computation load r. We refer to L^*_{dc} as the *minimum communication load*.

To obtain similar expressions for the communication load as in [3], we normalize the communication load by the total number of intermediate output bits ($=WNT$). We consider the first scenario now, where each reduce function is computed exactly at one node.

Theorem 3 ([3])**.** *The minimum communication load L^*_{dc} incurred by a distributed computing system of K nodes for a given computation load r, where every reduce function is computed at exactly one node and each node computes $\frac{W}{K}$ reduce functions, is bounded as*

$$\frac{L^*_{dc}}{WNT} \geq \frac{1}{r}\Big(1 - \frac{r}{K}\Big). \tag{37}$$

Proof. We resort to two of the three steps of Section 1.2 to complete this proof. The symmetrization step, which involves averaging over demand configurations, is not applicable in the present setting because the definition of L^*_{dc} involves minimization over the reduce function assignment as well.

Applying Theorem 1: Let $\mathcal{M} = (\mathcal{M}_1, \ldots, \mathcal{M}_K)$ denote a given map function assignment to the nodes, where $\mathcal{M}_i \subset [N]$. Let $L_{\mathcal{M}}$ denote the communication load associated with the map function assignment $\mathcal{M}$. We will prove that

$$\frac{L_{\mathcal{M}}}{WNT} \geq \sum_{j=1}^{K} \frac{\tilde{a}^j_{\mathcal{M}}}{N} \frac{K-j}{Kj},$$

where $\tilde{a}^j_{\mathcal{M}}$ denotes the number of files which are mapped at exactly j nodes in $[K]$. It is easy to see that $\sum_{j=1}^{K} \tilde{a}^j_{\mathcal{M}} = N$ and $\sum_{j=1}^{K} j\tilde{a}^j_{\mathcal{M}} = rN$. We will apply Theorem 1 to this setting. Recall that each reduce function is computed exactly at one node in our present setup. To apply Theorem 1, we need to ascertain the quantities a^Q_P for P, Q being disjoint subsets of $[K]$. To do this, we first denote by $\tilde{a}^Q$ the number of files whose intermediate outputs are demanded by some node k and available exclusively in the nodes of Q. Note that $\tilde{a}^Q$ is the same for any $k \in [K] \setminus Q$, as each node demands intermediate outputs of *all* the files that are not mapped at the node itself.

As the number of reduce functions assigned to node k is $\frac{W}{K}$ (as each reduce function is computed at exactly one node) and each intermediate output is T bits, the number of intermediate output bits which are demanded by any node k and available exclusively in the nodes of Q are $\frac{WT}{K}\tilde{a}^Q$. Thus, for any $Q \subset [K]$, the quantities a^Q_P in Theorem 1 are given as follows.

$$a^Q_P = \begin{cases} \frac{WT}{K}\tilde{a}^Q & \text{if } P = \{k\} \text{ for some } k \in [K] \text{ such that } k \notin Q \\ 0 & \text{otherwise} \end{cases}$$

Further note that $\sum_{Q \subset [K]:|Q|=j} \tilde{a}^Q = \tilde{a}^j_{\mathcal{M}}$ by definition of $\tilde{a}^j_{\mathcal{M}}$. Using these and applying Theorem 1 with the normalization factor WNT, we have the following inequalities.

$$\begin{aligned}
\frac{L_{\mathcal{M}}}{WNT} &\geq \frac{1}{WNT} \sum_{k=1}^{K} \sum_{Q \subset [K]\setminus\{k\}} \frac{1}{|Q|} \tilde{a}^Q \frac{WT}{K} \\
&= \frac{1}{KN} \sum_{j=1}^{K} \sum_{Q \subset [K]:|Q|=j} \sum_{k \in [K]\setminus Q} \frac{1}{j} \tilde{a}^Q \\
&= \frac{1}{KN} \sum_{j=1}^{K} \sum_{Q \subset [K]:|Q|=j} \frac{K-j}{j} \tilde{a}^Q \\
&= \frac{1}{KN} \sum_{j=1}^{K} \frac{K-j}{j} \Bigg(\sum_{Q \subset [K]:|Q|=j} \tilde{a}^Q \Bigg) \\
&= \frac{1}{KN} \sum_{j=1}^{K} \frac{K-j}{j} \tilde{a}^j_{\mathcal{M}}.
\end{aligned}$$

Refining the bound using convexity and setting constraints: Using definition of L^*_{dc}, noting that $\frac{K-j}{j}$ is a convex decreasing function of j and that $\frac{\sum_{j=1}^{K}\tilde{a}^j_{\mathcal{M}}}{N}=1$, we have that

$$\begin{aligned}\frac{L^*_{dc}}{WNT} &\geq \frac{1}{K}\frac{K-\sum_{j=1}^{K}\frac{j\tilde{a}^j_{\mathcal{M}}}{N}}{\sum_{j=1}^{K}\frac{j\tilde{a}^j_{\mathcal{M}}}{N}} \\ &= \frac{1}{K}\frac{K-r}{r}=\frac{1}{r}\Big(1-\frac{r}{K}\Big).\end{aligned}$$

□

Now, we consider the case in which each reduce function has to be computed at s nodes. The total number of reduce functions is assumed to be W. In addition, the following assumption is made to keep the problem formulation symmetric with respect to reduce functions: every possible s sized subset of K nodes is assigned $\frac{W}{\binom{K}{s}}$ reduce functions (we assume $\binom{K}{s}$ divides W). As in the previous case, we will denote the communication load for a given map function assignment by $L_{\mathcal{M}}(s)$ and the optimal communication load with computation load r by $L^*_{dc}(s)$. We will prove the following result which gives a lower bound on $L_{\mathcal{M}}(s)$.

Proposition 2 ([3])**.** *The communication load corresponding to a map function assignment $\mathcal{M}$ when each reduce function has to be computed at s nodes is lower bounded as*

$$\frac{L_{\mathcal{M}}(s)}{WNT}\geq\sum_{j=1}^{K}\frac{\tilde{a}^j_{\mathcal{M}}}{N}\sum_{l=\max(0,s-j)}^{\min(K-j,s)}\frac{\binom{K-j}{l}\binom{j}{s-l}}{\binom{K}{s}}\frac{l}{l+j-1}. \tag{38}$$

Proof. As before, we will denote by $\tilde{a}^Q$ the number of files whose map function outputs are available exclusively in the nodes of Q. Furthermore, we will denote the number of intermediate output bits which are demanded exclusively by the nodes in P and available exclusively in the nodes of Q by b^Q_P. Then, applying Theorem 1, the lower bound on the communication load in terms of $\{b^Q_P\}$ is given by

$$\frac{L_{\mathcal{M}}(s)}{WNT}\quad\geq\quad\frac{1}{WNT}\sum_{P\subset[K]}\sum_{Q\subset[K]\setminus P}\frac{|P|}{|P|+|Q|-1}b^Q_P.$$

We first interchange the above summation order and consider all sets Q with $|Q|=j$ and all sets P such that $|P|=l$. For $|Q|=j$, we need to count the subsets of size s, which form a subset of $P\cup Q$. Thus, for a fixed j, we can see that the range of l can vary from $\max(0,s-j)$ to $\min(K-j,s)$. For a given subset P of size l, the number of s sized subsets which are contained within $P\cup Q$ and contain P are $\binom{j}{s-l}$. Therefore, the number of intermediate output bits demanded exclusively by the nodes in P and available exclusively in Q, b^Q_P, is given by $b^Q_P=\tilde{a}^QWT\frac{\binom{j}{s-l}}{\binom{K}{s}}$. This is because each of the s-sized subset has to reduce $\tilde{a}^Q\frac{W}{\binom{K}{s}}$ functions. Using this relation, the above inequality can be rewritten as follows.

$$\begin{aligned}\frac{L_{\mathcal{M}}(s)}{WNT} &\geq \sum_{j=1}^{K}\frac{1}{N}\left(\sum_{Q\subset[K]:|Q|=j}\tilde{a}^Q\right)\sum_{l=\max(0,s-j)}^{\min(K-j,s)}\frac{l}{l+j-1}\left(\sum_{P\subset[K]\setminus Q:|P|=l}\frac{\binom{j}{s-l}}{\binom{K}{s}}\right) && (39)\\ &= \sum_{j=1}^{K}\frac{\tilde{a}^j_{\mathcal{M}}}{N}\sum_{l=\max(0,s-j)}^{\min(K-j,s)}\frac{l}{l+j-1}\frac{\binom{K-j}{l}\binom{j}{s-l}}{\binom{K}{s}}, && (40)\end{aligned}$$

where (40) follows as $\tilde{a}^j_{\mathcal{M}}=\sum_{Q\subset[K]:|Q|=j}\tilde{a}^Q$. This completes the proof. □

The above lemma along with certain convexity arguments resulting from the constraints imposed by the computation load can be used to prove the lower bound on $L^*_{dc}(s)$. The interested reader is referred to the converse proof of Theorem 2 in [3] for the same.

5. Relation to Index Coding Lower Bound

We now consider the "centralized" version of the data exchange problem, where one of the nodes has a copy of all the information bits and is the lone transmitter in the system. We will use the index 0 for this server node, and assume that there are K other nodes in the system, with index set $[K]$, acting as clients. In terms of Definition 1, this system is composed of $K+1$ nodes $\{0\} \cup [K]$, the demand D_0 of the server is empty, while the demands D_i and the contents C_i of all the clients are subsets of the contents of the server, i.e., $C_i, D_i \subset C_0$ for all $i \in [K]$. Without loss of generality, we assume that only the server performs all the transmissions as any coded bit that can be generated by any of the client nodes can be generated at the server itself. Clearly, this is an index coding problem [26] with K clients or receivers, the demand of the ith receiver is D_i, and its *side information* is C_i. When applied to this scenario, our main result Theorem 1 therefore provides a lower bound on the index coding communication cost.

The maximum acyclic induced subgraph (MAIS) and its generalization, which is known as the generalized independence number or the α-bound, are well-known lower bounds in index coding [9,26]. In this section, we describe the relation between the α-bound of index coding and the centralized version of Theorem 1. We show that the latter is in general weaker, and identify the scenarios when these two bounds are identical. We then use these observations to explain why Theorem 1 cannot provide a tight lower bound for the centralized data shuffling problem [6].

Let us first apply Theorem 1 to the centralized data exchange problem. As node 0 contains all the information bits and its demand is empty, we have $a_P^{Q'} = 0$ if $0 \notin Q'$ or $0 \in P$. Using $Q = Q' \setminus \{0\}$ and defining the variable $c_P^Q = a_P^{Q \cup \{0\}} = a_P^{Q'}$, we obtain

Theorem 4. *The centralized version of our main result Theorem 1 is*

$$\begin{aligned} L^* &\geq \sum_{P \subset [K]} \sum_{\substack{Q' \subset \{0\} \cup [K] \\ 0 \in Q', P \cap Q' = \emptyset}} \frac{|P|}{|P| + |Q'| - 1} a_P^{Q'} \\ &= \sum_{P \subset [K]} \sum_{Q \subset [K] \setminus P} \frac{|P|}{|P| + |Q|} c_P^Q. \end{aligned}$$

Note that it is possible to have $c_P^Q = a_P^{Q \cup \{0\}} > 0$ when $Q = \emptyset$.

In Section 5.1, we express the generalized independence number α in terms of the parameters c_P^Q, and in Section 5.2, we identify the relation between our lower bound Theorem 4 and the index coding lower bound α.

5.1. The Generalized Independence Number Bound

Let $\gamma = (\gamma_1, \ldots, \gamma_K)$ be any permutation of $[K]$, where γ_i is the ith coordinate of the permutation. Applying similar ideas as in the proof of Theorem 1 to the centralized scenario, we obtain the following lower bound on L^*. This lower bound considers the nodes in the order $\gamma_1, \ldots, \gamma_K$, and for each node in this sequence it counts the number of bits that are demanded by this node which are neither demanded by and nor available as side information in any of the earlier nodes.

Proposition 3. *For any permutation γ of $[K]$,*

$$L^* \geq \sum_{i=1}^{K} \sum_{\substack{P \subset \{\gamma_i, \dots, \gamma_K\} \\ \gamma_i \in P}} \sum_{Q \subset \{\gamma_{i+1}, \dots, \gamma_K\}} c_P^Q. \tag{41}$$

Proof. See Appendix D. □

A direct consequence of Proposition 3 is

$$L^* \geq \max_{\gamma} \sum_{i=1}^{K} \sum_{\substack{P \subset \{\gamma_i, \dots, \gamma_K\} \\ \gamma_i \in P}} \sum_{Q \subset \{\gamma_{i+1}, \dots, \gamma_K\}} c_P^Q \tag{42}$$

where the maximization is over all possible permutations on $[K]$.

We now recall the definition of the generalized independence number [9]. Denote the collection of the c_P^Q information bits available exclusively at the nodes $Q \cup \{0\}$ and demanded exclusively by the nodes P as $\{w_{P,m}^Q \,:\, m = 1, \dots, c_P^Q\}$. Therefore, the set of all the information bits present in the system is

$$B = \bigcup_{P \subset [K]} \bigcup_{Q \subset [K] \setminus P} \left\{ w_{P,m}^Q \,:\, m = 1, \dots, c_P^Q \right\}.$$

Note that each bit is identified by a triple (P, Q, m).

Definition 2. *A subset $\mathcal{H}$ of B is a generalized independent set if and only if every subset $\mathcal{I} \subset \mathcal{H}$ satisfies the following:*

- *there exists a node $k \in [K]$ and an information bit in $\mathcal{I}$ such that this information bit is demanded by k (and possibly some other nodes), and none of the other bits in $\mathcal{I}$ are available as side information at k.*

The generalized independence number α is the size of the largest generalized independent set.

We next show that the lower bound in (42) is in fact equal to the generalized independence number α of this index coding problem.

Theorem 5. *The generalized independence number α satisfies*

$$\alpha = \max_{\gamma} \sum_{i=1}^{K} \sum_{\substack{P \subset \{\gamma_i, \dots, \gamma_K\} \\ \gamma_i \in P}} \sum_{Q \subset \{\gamma_{i+1}, \dots, \gamma_K\}} c_P^Q, \tag{43}$$

where the maximization is over all $K!$ permutations of $[K]$.

Proof. See Appendix E. □

5.2. Relation to the Index Coding Lower Bound

Proposition 3 serves as the platform for comparing Theorem 4 and the α-bound. While α equals the *maximum* value of the bound in Proposition 3 over all permutations on $[K]$, our bound in Theorem 4 equals the *average* value of the lower bound given in Proposition 3 over all permutations on $[K]$. We will show this relation between Theorem 4 and Proposition 3 now.

Taking the average of the right hand side of (41) with respect to all γ, we obtain

$$\frac{1}{K!}\sum_{\gamma}\sum_{i=1}^{K}\sum_{\substack{P\subset\{\gamma_i,\dots,\gamma_K\}\\ \gamma_i\in P}}\sum_{Q\subset\{\gamma_{i+1},\dots,\gamma_K\}} c_P^Q.$$

For each choice of $P, Q \subset [K]$ with $P \cap Q = \emptyset$, we now count the number of times c_P^Q appears in this sum. For a given γ, the inner summations include the term c_P^Q if and only if the following holds:

$$\gamma_i \in P, \text{ where } i = \min\{j \in [K] \,:\, \gamma_j \in P \cup Q\},$$

i.e., if we consider the elements $\gamma_1, \dots, \gamma_K$ in that order, the first element from $P \cup Q$ to be observed in this sequence belongs to P. Thus, for a given pair P, Q the probability that a permutation γ chosen uniformly at random includes the term c_P^Q in the inner summation is $|P|/(|P| + |Q|)$. Therefore, the average of the lower bound in Proposition 3 over all possible γ is

$$\sum_{P\subset[K]}\sum_{\substack{Q\subset[K]\\ P\cap Q=\emptyset}}\frac{|P|}{|P|+|Q|}c_P^Q,$$

which is exactly the bound in Theorem 4.

As the bound in Theorem 4 is obtained by averaging over all γ, instead of maximizing over all γ, we conclude that this is in general weaker than the α-bound of index coding. The two bounds are equal if and only if the bound in Proposition 3 has the same value for every permutation γ.

Although weaker in general, we note that the bound of Theorem 4 is easier to use than the α-bound. As demonstrated by (2), in order to use Theorem 4, we only need to know, for each information bit, the number of nodes that contain this bit and the number of nodes that demand this bit. In comparison, this information is insufficient to evaluate the α-bound, which also requires the identities of these nodes.

5.3. On the Tightness of Theorem 4

We now consider the class of *unicast* problems, i.e., problems where each bit is demanded by exactly one of the nodes. For this class of problems, we characterize when Theorem 4 yields a tight bound.

Theorem 6. *For unicast problems the bound in Theorem 4 equals L^* if and only if every $S \subset [K]$ with $|S| \geq 2$ satisfies the following, $c_{\{k\}}^{S\setminus k} = c_{\{k'\}}^{S\setminus k'}$ for every $k, k' \in S$.*

Proof. See Appendix F. When the lower bound of Theorem 4 is tight, the clique-covering based index coding scheme (see in [26,27]) yields the optimal communication cost. □

Our main result in Theorem 1, or equivalently, Theorem 4, does not provide a tight lower bound for centralized data shuffling problem [6], because this problem involves scenarios that do not satisfy the tightness condition of Theorem 6. For instance, consider the simple canonical data shuffling setting, where the system has exactly K files, all of equal size F bits, and each node stores exactly one of these files, i.e., the entirety of the contents of the kth node C_k is the kth file. Here, $|C_k| = F$ for all $k \in [K]$, and $C_i \cap C_j = \emptyset$ for all $i \neq j$. Assume that the shuffling problem is to move the file C_{k+1} to node k, i.e., $D_k = C_{k+1}$, where we consider the index $K + 1$ to be equal to 1. This is a worst-case demand for data shuffling incurring the largest possible communication cost. For this set of demands, we have $c_k^{\{k+1\}} = F$ for all $k \in [K]$, and $c_k^Q = 0$ for all other choices of k, Q. In particular, $c_{k+1}^{\{k\}} = 0 \neq c_k^{\{k+1\}}$. Clearly, the condition in Theorem 6 does not hold for $S = \{k, k + 1\}$.

Therefore, our lower bound is strictly less than L^* for this data shuffling problem, and therefore is not tight.

6. Relationship to Other Index Coding Settings

We now comment on the application of our data exchange bound to a couple of other important index coding settings known in literature, (a) *distributed index coding* studied in [10] (which is equivalent to the *cooperative multi-sender index coding* setting considered in [11]), and (b) *embedded index coding*, presented in [12].

6.1. Distributed Index Coding

In [10], the authors consider a generalization of the single-server index coding problem (which we studied in Section 5) called *distributed index coding*. The specific setting in [10] is as follows. There are n messages denoted by $x_j : j \in [n]$, where $x_j \in \{0,1\}^{t_j}$ (for some positive integer t_j). There is a corresponding set of n receivers indexed by $[n]$. The receiver $j \in [n]$ contains as side-information the subset of messages indexed by $A_j \subset [n]$ (i.e., receiver j knows $\{x_i : i \in A_j\}$) and demands the message x_j. There are $2^n - 1$ servers in the system, indexed by the sets $\mathcal{J} = \{J : J \subset [n], J \neq \emptyset\}$. The server J contains the messages $\{x_i : i \in J\}$. The servers do not demand any messages and are responsible only for transmissions that satisfy the receivers. The server J is connected to the n receivers via a broadcast link with capacity C_J bits. In order to satisfy the demands, each server J sends a message $y_J \in \{0,1\}^{s_J}$ to all the receivers, where s_J is some positive integer.

Definition 3 ([10])**.** *The rate-capacity tuple $((R_j : j \in [n]), (C_J : J \in \mathcal{J}))$ is said to be achievable if there exists some positive integer r such that $t_j \geq rR_j, \forall j$ and $s_J \leq rC_J, \forall J$, and there exists valid encoding functions (encoding the messages of lengths $(t_j : j \in [n])$ into codewords of lengths $(s_J : J \in \mathcal{J})$) and decoding functions, such that all receivers can decode their respective demands.*

Slightly abusing Definition 2, for some $T \subset [n]$, we call a set $S \subset T$ of message indices as a generalized independent set of T, if for every subset $S' \subset S$, there is some $j \in S'$ such that $A_j \cap (S' \setminus j) = \emptyset$.

Let $((R_j : j \in [n]), (C_J : J \in \mathcal{J}))$ be an achievable rate-capacity tuple. For any non-empty subset $T \subset [n]$, let S_T be a generalized independent set of T. In Corollary 2 of [10], it is shown that

$$\sum_{j \in S_T} R_j \leq \sum_{J : J \cap T \neq \emptyset} C_J. \tag{44}$$

Remark 6. *The above bound in (44) is given in [10] using the terminology of the side-information graph defining the index coding problem and its acyclic induced subgraphs. However, we have used generalized independent sets to state the same bound. The reader can easily confirm that the acyclic induced subgraph of the side-information graph as defined in [10] is the same as a generalized independent set we have used in this work. Therefore, (44) is the same as the bound in Corollary 2 of [10].*

Let

$$S_{max} = arg \max_{S} \sum_{j \in S} t_j,$$

where the maximization is over all generalized independent sets S of $[n]$.

Then, we have by (44),

$$\sum_{j \in S_{max}} R_j \leq \sum_{J \in \mathcal{J}} C_J. \tag{45}$$

In order to relate the bound in (44) with our data exchange bound, we fix $R_j = t_j, \forall j \in [n]$. This means that we should have $r = 1$ in Definition 3. For these parameters, let $s_J^* : J \in \mathcal{J}$

be a choice of integers $s_J : J \in \mathcal{J}$ such that the rate-capacity tuple $((R_j = t_j : j \in [n]), (s_J : J \in \mathcal{J}))$ is achievable and $\sum_{J\in\mathcal{J}} s_J$ is minimized. Note that such integers $s_J^* : J \in \mathcal{J}$ will exist as each index coding problem has at least one solution, namely, the trivial solution consisting of uncoded transmissions of $x_j : j \in [n]$.

Then, applying (45), we have

$$\sum_{j\in S_{max}} t_j \leq \sum_{J\in\mathcal{J}} s_J^*. \tag{46}$$

Note that $\sum_{J\in\mathcal{J}} s_J^*$ is exactly the minimum number of bits to be communicated by the servers for satisfying receiver demands.

For $Q \subset [n]$, define

$$f_j^Q \triangleq \begin{cases} 1 & \text{if } j \in (\cap_{k\in Q} A_k) \setminus (\cup_{k'\in[n]\setminus Q} A_{k'}), \\ 0 & \text{otherwise.} \end{cases}$$

By arguments similar to that of the proof of Theorem 5, we can verify that

$$\sum_{j\in S_{max}} t_j = \max_{\gamma} \sum_{j=1}^{n} \sum_{Q\subset\{\gamma_{j+1},\ldots,\gamma_n\}} f_{\gamma_j}^Q t_{\gamma_j}, \tag{47}$$

where the maximization is over all possible permutations $\gamma = (\gamma_1, \ldots, \gamma_n)$ of $(1, \ldots, n)$. We thus have by (46) and (47),

$$\sum_{J\in\mathcal{J}} s_J^* \geq \max_{\gamma} \sum_{j=1}^{n} \sum_{Q\subset\{\gamma_{j+1},\ldots,\gamma_n\}} f_{\gamma_j}^Q t_{\gamma_j}. \tag{48}$$

Finally, we apply our data exchange bound in Theorem 1 to the distributed index coding setting. To do this, we first observe that if we replaced all servers by a single "virtual" central server containing all the messages, $x_j : j \in [n]$, then $\sum_{J\in\mathcal{J}} s_J^*$ is the minimum number of bits to be transmitted by this virtual central server to satisfy the receiver demands. Any lower bound on the communication cost for this transformed setting with the virtual server will thus continue to apply for the original distributed setting with messages of length $t_j : j \in [n]$. Now, utilizing the centralized version of Theorem 1 shown in Theorem 4 and by the discussion in Section 5.2, we get

$$\sum_{J\in\mathcal{J}} s_J^* \geq \frac{1}{n!} \sum_{\gamma} \sum_{j=1}^{n} \sum_{Q\subset\{\gamma_{j+1},\ldots,\gamma_n\}} f_{\gamma_j}^Q t_{\gamma_j}. \tag{49}$$

Therefore, we see that the generalized independent set based bound in (48) is in general better than (49), as (48) involves a maximization over all permutations γ, while (49) involves the average.

6.2. Embedded Index Coding

We now consider the embedded index coding problem, introduced in [12], motivated by device-to-device communications. The embedded index coding setting consists of a set of m data blocks (each a binary vector of length t) distributed across a set of n nodes. Each node stores (as side information) a subset of the data blocks and demands another subset which it already does not have. This setting is different from [26,27] or distributed index coding [10], as there are no dedicated servers by default here. Each node transmits a codeword obtained by encoding its data blocks, and each demanded data block at any node is decoded from the codewords obtained from other nodes and the side information at the node itself. An *embedded index code* consists of a collection of such encoding functions and decoding functions at the nodes, such that all demanded blocks are decoded at the respective nodes. The *communication cost* of embedded index coding is the total number of bits transmitted between the nodes to satisfy the node demands. The work [12] generalizes the notion of *minrank* [26] of single-server index codes to define the optimal length of linear embedded index codes. Further, the authors also present heuristic constructions for general and specialized linear codes which have some nice properties.

As the embedded index coding problem clearly has a direct mapping with the data exchange problem considered in the present work, we can apply our data exchange bound directly to obtain a new lower bound for the communication cost of embedded index coding. The expression of this bound would be in the same form (up to only the change in notation) as Theorem 1 itself. As our bound holds in the information-theoretic sense, it would apply to not just the linear codes considered in [12] but nonlinear embedded index codes as well.

7. Conclusions

We have presented an information theoretic converse result for a generic data exchange problem, where the terminals contain some data in their local storage and want other data available at the local storage of other nodes. As a number of recently studied multi-terminal communication problems fall under this setting, we have used our general converse to obtain converses in many such settings, thus recovering many existing results and presenting some new results as well. Using a connection with index coding, we also presented some ideas on why and when our data exchange based converse can be loose in the index coding setting. It would be quite interesting to see if our converse result can be tightened further while still retaining a closed form expression, so as to cover all known bounds for any existing setting that can be modeled in the data exchange framework. A lower bound for the communication load in a generic data exchange setting in the presence of coded storage bits would also be a prospective direction for future research in this area.

Author Contributions: Conceptualization, P.K. and L.N.; formal analysis, P.K., L.N., V.L.; writing—original draft preparation, P.K., L.N., V.L.; writing—review and editing, P.K., L.N., V.L.; funding acquisition, P.K., L.N., V.L. All authors have read and agreed to the published version of the manuscript.

Funding: Prasad Krishnan was supported by Science and Engineering Research Board (SERB), Government of India via grants CRG/2019/005572 and MTR/2017/000827. Lakshmi Natarajan was supported by Science and Engineering Research Board (SERB-DST), Government of India (via projects MTR/2019/001454 and DST/INSPIRE/04/2015/002094).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Acknowledgments: The authors are thankful to the anonymous reviewers and the academic editor for their careful reading of the manuscript and comments that helped improved the quality of the manuscript.

Conflicts of Interest: The authors declare no conflict of interest.

Appendix A. Proof of Theorem 1

We assume that all the bits in the collection B as in Definition 1 are i.i.d uniformly distributed on $\{0,1\}$. For a given communication scheme for the given data exchange problem, let $X_i \triangleq \phi_i(C_i)$ represent the codeword transmitted by node i. For a subset $S \subset [K]$, let $X_S \triangleq \cup_{i \in S} X_i$. Furthermore, let $Y_S = \bigcup_{i \in S}(D_i \cup C_i)$. We first prove the following claim.

Claim A1. *For any $S \subset [K]$,*

$$H(X_S|Y_{\overline{S}}) \geq \sum_{P \subset S} \sum_{Q \subset S \setminus P} \frac{|P|}{|P|+|Q|-1} a_P^Q, \tag{A1}$$

where $\overline{S} = [K] \setminus S$.

Applying $S = [K]$ to the above claim then gives Theorem 1, as $L^* \geq H(X_{[K]})$.

Now, we prove Claim A1. For this, we use induction on $|S|$.

We take the base case to be $|S| = 2$, as for $|S| = 1$ the problem of data exchange is not well defined. Let $S = \{1,2\}$ without loss of generality. Then, the LHS of (A1) gives

$$\begin{aligned} H(X_1, X_2|Y_{\overline{S}}) &\geq H(X_1|Y_{\overline{S}}) + H(X_2|Y_{\overline{S}}, X_1) \\ &\geq H(X_1|Y_{\overline{S}}, C_2) + H(X_2|Y_{\overline{S}}, C_1) && \text{(A2)} \\ &= H(X_1, D_2|Y_{\overline{S}}, C_2) + H(X_2, D_1|Y_{\overline{S}}, C_1) && \text{(A3)} \\ &\geq H(D_2|Y_{\overline{S}}, C_2) + H(D_1|Y_{\overline{S}}, C_1) && \text{(A4)} \\ &\geq a_{\{2\}}^{\{1\}} + a_{\{1\}}^{\{2\}}, && \text{(A5)} \end{aligned}$$

where (A2) follows as conditioning reduces entropy and $H(X_1|C_1) = 0$, (A3) is true as $H(D_2|Y_{\overline{S}}, C_2, X_1) = 0$ and $H(D_1|Y_{\overline{S}}, C_1, X_2) = 0$. This proves the base case.

We now assume that the statement is true for $|S| = t-1$, and prove that it holds for $|S| = t$. We have the LHS of (A1) satisfying the following relationships for $|S| = t$.

$$\begin{aligned} H(X_S|Y_{\overline{S}}) &= \frac{1}{t}\sum_{k\in S}\Big(H(X_{S\setminus k}|Y_{\overline{S}}, X_k) + H(X_k|Y_{\overline{S}})\Big) \\ &\geq \frac{1}{t}\left(\sum_{k\in S} H(X_{S\setminus k}|Y_{\overline{S}}, C_k)\right) + \frac{1}{t}H(X_S|Y_{\overline{S}}) && \text{(A6)} \\ &\geq \frac{1}{t-1}\sum_{k\in S} H(X_{S\setminus k}|Y_{\overline{S}}, C_k) \\ &= \frac{1}{t-1}\sum_{k\in S} H(X_{S\setminus k}, D_k|Y_{\overline{S}}, C_k) && \text{(A7)} \\ &= \tfrac{1}{t-1}\textstyle\sum_{k\in S}\Big(H(D_k|C_k, Y_{\overline{S}}) + H(X_{S\setminus k}|Y_{\overline{S\setminus k}})\Big), && \text{(A8)} \end{aligned}$$

where (A6) follows because $H(X_k|C_k) = 0$. In (A7), we introduce D_k freely, because

$$\begin{aligned} H(D_k|X_{S\setminus k}, Y_{\overline{S}}, C_k) &\leq H(D_k|X_{S\setminus k}, C_{\overline{S}}, C_k) \\ &\leq H(D_k|X_{S\setminus k}, X_{\overline{S}}, C_k) \\ &= 0, \end{aligned}$$

where the last two statements follow because $H(X_{\overline{S}}|C_{\overline{S}}) = 0$ and from the decoding condition, respectively. We now interpret the two terms of (A8). For the first term, we have

$$\begin{aligned} \sum_{k\in S} H(D_k|C_k, Y_{\overline{S}}) &= \sum_{k\in S}\sum_{P'\subset S\setminus k}\sum_{Q\subset S\setminus(P'\cup k)} a_{k\cup P'}^{Q} \\ &= \sum_{P\subset S}\sum_{Q\subset S\setminus P} |P| a_P^Q, && \text{(A9)} \end{aligned}$$

where the last statement follows by noting that for a fixed choice of P, Q we have $|P|$ choices for (k, P') such that $P' \cup k = P$.

Now, using the induction hypothesis for the last term of (A8),

$$\sum_{k\in S} H(X_{S\setminus k}|Y_{\overline{S\setminus k}})$$
$$\geq \sum_{k\in S}\sum_{P\subset S\setminus k}\sum_{Q\subset S\setminus(P\cup k)} \frac{|P|}{|P|+|Q|-1} a_P^Q, \tag{A10}$$
$$= \sum_{P\subset S}\sum_{Q\subset S\setminus P} (t-|P|-|Q|)\frac{|P|}{|P|+|Q|-1} a_P^Q \tag{A11}$$

where the above follows by noting that for a fixed choice of disjoint subsets P, Q of S, we have $|S|-|P|-|Q|$ choices for k such that $P\subset S\setminus k$ and $Q\subset S\setminus(P\cup k)$.

Using (A9) and (A11) we have

$$\text{RHS of (A8)}$$
$$\geq \frac{1}{t-1}\sum_{P\subset S}\sum_{Q\subset S\setminus P}|P|\left(1+\frac{t-|P|-|Q|}{|P|+|Q|-1}\right)a_P^Q$$
$$= \sum_{P\subset S}\sum_{Q\subset S\setminus P}\frac{|P|}{|P|+|Q|-1}a_P^Q,$$

thus proving Claim A1, which also concludes the proof of the theorem.

Appendix B. Proof of (11)

We proceed according to the three steps in Section 1.2, however with some important variations that are required to prove (11).

Applying Theorem 1 and symmetrizing: As in the proof of Theorem 2, we use the index 0 to represent the server. Consider that, for the given placement scheme ζ, the delivery scheme is designed so that the optimal communication load $L_c^*(N_u,\zeta)$ is achieved. Let $\mathcal{A} = \binom{[K]}{N_u}$ be the set of all N_u-sized subsets of clients. Consider the coded caching subproblem induced by the server and a set $A\in\mathcal{A}$ of clients. Consider some demand vector $\boldsymbol{d} = (d_1,\ldots,d_K)$ such that the demands of the clients in A are distinct, i.e., $d_i\neq d_j$ for $i,j\in A, i\neq j$.

Let Ω consist of the $N-1$ cyclic permutations of $(1,\ldots,N)$ which are N-cycles, along with the identity permutation. For $\sigma\in\Omega$, let $\sigma(\boldsymbol{d})$ denote the demand vector in which the ith component is exactly the value obtained by applying the permutation σ on the ith component of $\boldsymbol{d}$, i.e., $\sigma(\boldsymbol{d})_i = \sigma(d_i)$.

Clearly, for each $\sigma\in\Omega$, we have $L_c^*(N_u,\zeta)\geq L_A^*(\sigma(\boldsymbol{d}))$, where $L_A^*(\sigma(\boldsymbol{d}))$ is the optimal communication load for this subproblem with demands $\sigma(\boldsymbol{d})$ with respect to the placement ζ.

Now, for each $\sigma\in\Omega$, following similar steps as in proof of Theorem 2, we can use our data exchange bound in Theorem 1 to obtain

$$L_c^*(N_u,\zeta)\geq L_A^*(\sigma(\boldsymbol{d}))\geq\sum_{k\in A}\sum_{Q_1\subset A\setminus k}\frac{1}{1+|Q_1|}f_k^{Q_1}(\sigma(\boldsymbol{d})), \tag{A12}$$

where $f_k^{Q_1}(\sigma(\boldsymbol{d}))$ is the number of bits of the demanded file $W_{\sigma(\boldsymbol{d})_k}$ of the client k available at the nodes $Q_1\subset A$ and no other nodes in A. Let a^Q be the number of all bits (corresponding to all N files) stored exclusively in clients $Q\cup\{0\}\subset[K]\cup\{0\}$. Then, by the structure of the demand vectors in the set $\mathcal{D}=\{\sigma(\boldsymbol{d}):\sigma\in\Omega\}$, we see that

$$\sum_{\sigma\in\Omega} f_k^{Q_1}(\sigma(\boldsymbol{d})) = \sum_{Q_2\subset[K]\setminus A} a^{Q_1\cup Q_2}.$$

By averaging (A12) across the N demand vectors in $\{\sigma(\boldsymbol{d}) : \sigma \in \Omega\}$, we get

$$L_c^*(N_u,\zeta) \geq \frac{1}{N}\sum_{k\in A}\sum_{Q_1\subset A\setminus k}\sum_{Q_2\subset[K]\setminus A}\frac{1}{1+|Q_1|}a^{Q_1\cup Q_2}. \tag{A13}$$

For each $A \in \mathcal{A}$, the bound (A13) holds. Averaging these bounds for all $A \in \mathcal{A}$, we obtain

$$L_c^*(N_u,\zeta) \geq \frac{1}{N\binom{K}{N_u}}\sum_{A\in\mathcal{A}}\sum_{k\in A}\sum_{Q_1\subset A\setminus k}\sum_{Q_2\subset[K]\setminus A}\frac{1}{1+|Q_1|}a^{Q_1\cup Q_2}. \tag{A14}$$

In the above summation, for any given A, for fixed $Q_1 \subset A, Q_2 \subset [K] \setminus A$, the variable k takes values from $A \setminus Q_1$. Thus, we have

$$\begin{aligned} L_c^*(N_u,\zeta) &\geq \frac{1}{N\binom{K}{N_u}}\sum_{A\in\mathcal{A}}\sum_{Q_1\subset A}\sum_{Q_2\subset[K]\setminus A}\left(\frac{N_u-|Q_1|}{1+|Q_1|}\right)a^{Q_1\cup Q_2} && \text{(A15)}\\ &= \frac{1}{N\binom{K}{N_u}}\sum_{q=0}^{N_u}\sum_{Q\subset[K]}\sum_{A\in\mathcal{A}:|Q\cap A|=q}\left(\frac{N_u-q}{1+q}\right)a^{Q}. && \text{(A16)}\end{aligned}$$

For some $Q \subset [K]$, to obtain an $A \in \mathcal{A}$ such that $|A \cap Q| = q$, we have to choose q elements from Q and $N_u - q$ elements from outside Q. Thus, we have $|\{A \in \mathcal{A} : |Q \cap A| = q\}| = \binom{|Q|}{q}\binom{K-|Q|}{N_u-q}$. Thus, we have

$$L_c^*(N_u,\zeta) \geq \frac{1}{N\binom{K}{N_u}}\sum_{q=0}^{N_u}\sum_{Q\subset[K]}\binom{|Q|}{q}\binom{K-|Q|}{N_u-q}\left(\frac{N_u-q}{1+q}\right)a^{Q}. \tag{A17}$$

Now

$$\begin{aligned}\binom{|Q|}{q}\binom{K-|Q|}{N_u-q}\left(\frac{N_u-q}{1+q}\right) &= \frac{|Q|!(K-|Q|)!(N_u-q)}{q!(|Q|-q)!(N_u-q)!(K-|Q|-N_u+q)!(1+q)} && \text{(A18)}\\ &= \binom{K}{N_u}\frac{\binom{N_u}{q+1}\binom{K-N_u}{|Q|-q}}{\binom{K}{|Q|}}. && \text{(A19)}\end{aligned}$$

Further, we have

$$\sum_{q=0}^{N_u}\binom{N_u}{q+1}\binom{K-N_u}{|Q|-q} = \sum_{q=1}^{N_u}\binom{N_u}{q}\binom{K-N_u}{|Q|-q+1} = \binom{K}{|Q|+1} - \binom{K-N_u}{|Q|+1}. \tag{A20}$$

Using (A20) and (A19) in (A17), we get

$$\begin{aligned} L_c^*(N_u,\zeta) &\geq \frac{1}{N}\sum_{Q\subset[K]}\frac{\binom{K}{|Q|+1}-\binom{K-N_u}{|Q|+1}}{\binom{K}{|Q|}}a^{Q}\\ &\geq \frac{1}{NF}\sum_{Q\subset[K]}g_{N_u}(|Q|)a^{Q}, && \text{(A21)}\end{aligned}$$

where g_{N_u} is the lower convex envelope as defined in Remark 2.

Using the constraints to revise (A21):

Observe that $\sum_{Q\subset[K]} a^Q = NF$. We assume without loss of generality that all caches are completely populated with the bits of the file library (as we are bounding the optimal load), and thus we have $\sum_{Q\subset[K]} |Q|a^Q = tNF$. By using these and applying Jensen's inequality to (A21), we have (11). This completes the proof.

Appendix C. Proof of (21)

The proof uses the same approach as in Appendix B, but takes into account the fact that there is no central server and the cache sizes are heterogeneous. The choice of the set of demand vectors used for symmetrization is the same as in Appendix B.

Applying Theorem 1 and symmetrizing: Let the sets $\mathcal{A}$ and $A \in \mathcal{A}$, the set of cyclic permutations Ω, demand vector $\boldsymbol{d}$, and optimal communication loads $L_c^*(N_u, \zeta)$ and $L_A^*(\sigma(\boldsymbol{d}))$ be defined as in Appendix B. Throughout this proof we will assume $N_u = \min\{N, K\}$. Applying our main result Theorem 1 to the subproblem induced by the demands of the subset of nodes A, we obtain

$$L_c^*(N_u,\zeta) \geq L_A^*(\sigma(\boldsymbol{d})) \geq \sum_{k\in A} \sum_{Q\subset[K]\setminus k} \frac{1}{|Q|} c_k^Q(\sigma(\boldsymbol{d})),$$

where $c_k^Q(\sigma(\boldsymbol{d}))$ is the number of bits of the file $W_{(\sigma(\boldsymbol{d}))_k}$ demanded by node k available exclusively in all nodes in Q and not available in $[K] \setminus Q$. Let a^Q be the total number of bits stored exclusively in the nodes Q. Then, considering the set of N demands $\{\sigma(\boldsymbol{d}) : \sigma \in \Omega\}$, we have

$$\sum_{\sigma\in\Omega} c_k^Q(\sigma(\boldsymbol{d})) = a^Q.$$

Averaging over all possible σ, we obtain

$$L_c^*(N_u,\zeta) \geq \frac{1}{N} \sum_{\sigma\in\Omega} L_A^*(\sigma(\boldsymbol{d})) \geq \frac{1}{N} \sum_{k\in A} \sum_{Q\subset[K]\setminus k} \frac{1}{|Q|} a^Q.$$

Again, averaging this inequality over all possible choices of $A \in \mathcal{A} = \binom{[K]}{N_u}$, we obtain

$$\begin{aligned} L_c^*(N_u,\zeta) &\geq \frac{1}{N\binom{K}{N_u}} \sum_{A\in\mathcal{A}} \sum_{\sigma\in\Omega} L_A^*(\sigma(\boldsymbol{d})) \\ &\geq \frac{1}{N\binom{K}{N_u}} \sum_{A\in\mathcal{A}} \sum_{k\in A} \sum_{Q\subset[K]\setminus k} \frac{1}{|Q|} a^Q. \end{aligned} \tag{A22}$$

In (A22), for a given choice of $Q \subset [K]$, the term $\frac{1}{|Q|} a^Q$ appears in the summation for every choice of (A, k) such that $k \notin Q$, $|A| = N_u$, $k \in A$. Therefore, the number of times $\frac{1}{|Q|} a^Q$ appears in (A22) is the number of such choices of (A, k) which is $(K - |Q|)\binom{K-1}{N_u-1}$. Hence, we arrive at

$$\begin{aligned} L_c^*(N_u,\zeta) &\geq \frac{1}{N\binom{K}{N_u}} \sum_{Q\subset[K]} \binom{K-1}{N_u-1} \frac{(K-|Q|)}{|Q|} a^Q \\ &= \frac{FN_u}{K} \sum_{Q\subset[K]} \frac{(K-|Q|)}{|Q|} \frac{a^Q}{NF}. \end{aligned} \tag{A23}$$

Using the constraints to revise (A23): We use the observations that $\frac{K-x}{x}$ is convex and decreasing in $x > 0$, $\sum_{Q\subset[K]} \frac{a^Q}{NF} = 1$, and $\sum_{Q\subset[K]} |Q| a^Q \leq \sum_{i=1}^t K_{T_i} \gamma_{T_i} NF$ since this is the total available cache memory across all nodes. Applying Jensen's inequality on (A23) using these constraints, we obtain

$$L_c^*(N_u,\zeta) \geq \frac{FN_u}{K} \frac{K - \sum_{i=1}^t K_{T_i}\gamma_{T_i}}{\sum_{i=1}^t K_{T_i}\gamma_{T_i}}.$$

This completes the proof.

Appendix D. Proof of Proposition 3

We will continue to use the notations used in the proof of Theorem 1. Let $k \in [K]$, and $S = \{\gamma_k, \ldots, \gamma_K\}$. Note that $\overline{S} = \{\gamma_1, \ldots, \gamma_{k-1}\}$. We will prove by induction on $|S| = K - k + 1$ that

$$H(X_0|Y_{\overline{S}}) \geq \sum_{i=k}^{K} \sum_{\substack{P \subset \{\gamma_i,\ldots,\gamma_K\} \\ \gamma_i \in P}} \sum_{Q \subset \{\gamma_{i+1},\ldots,\gamma_K\}} c_P^Q. \tag{A24}$$

Then, the result claimed in the proposition follows by using $S = \{\gamma_1, \ldots, \gamma_K\} = [K]$, i.e., $k = 1$. When $|S| = 1$, i.e., $k = K$ and $S = \{\gamma_K\}$, clearly (A24) is true, as

$$H(X_0|Y_{\{\gamma_1,\ldots,\gamma_{K-1}\}}) \geq a_{\{\gamma_K\}}^{\{0\}} = c_S^{\emptyset}.$$

Now, consider $S = \{\gamma_k, \ldots, \gamma_K\}$. The induction hypothesis is

$$H(X_0|Y_{\overline{S\setminus\gamma_k}}) \geq \sum_{i=k+1}^{K} \sum_{\substack{P \subset \{\gamma_i,\ldots,\gamma_K\} \\ \gamma_i \in P}} \sum_{Q \subset \{\gamma_{i+1},\ldots,\gamma_K\}} c_P^Q. \tag{A25}$$

Using the fact $H(D_{\gamma_k}|Y_{\overline{S}}, C_{\gamma_k}, X_0) \leq H(D_{\gamma_k}|C_{\gamma_k}, X_0) = 0$, we have

$$\begin{aligned}
H(X_0|Y_{\overline{S}}) &\geq H(X_0|Y_{\overline{S}}, C_{\gamma_k}) \\
&= H(X_0|Y_{\overline{S}}, C_{\gamma_k}) + H(D_{\gamma_k}|Y_{\overline{S}}, C_{\gamma_k}, X_0) \\
&= H(X_0, D_{\gamma_k}|Y_{\overline{S}}, C_{\gamma_k}) \\
&= H(D_{\gamma_k}|Y_{\overline{S}}, C_{\gamma_k}) + H(X_0|C_{\gamma_k}, D_{\gamma_k}, Y_{\overline{S}}) \\
&= \sum_{\substack{P \subset \{\gamma_k,\ldots,\gamma_K\} \\ \gamma_k \in P}} \sum_{Q \subset \{\gamma_{k+1},\ldots,\gamma_K\}} c_P^Q + H(X_0|Y_{\overline{S\setminus\gamma_k}})
\end{aligned} \tag{A26}$$

We observe that (A24) follows from (A25) and (A26).

Appendix E. Proof of Theorem 5

We prove this theorem by showing that α is both upper and lower bounded by the right hand side of (43).

Upper Bound: Assume that $\mathcal{H}$ is a largest generalized independent set. We will now identify a permutation $\pi = (\pi_1, \ldots, \pi_K)$ corresponding to $\mathcal{H}$. Let $\mathcal{I}_1 = \mathcal{H}$, and observe that as $\mathcal{I}_1$ is itself a subset of $\mathcal{H}$, it must contain an information bit, say $w_{P,m}^Q$ that is demanded by a node, say π_1, and none of the bits in $\mathcal{I}_1 \setminus \{w_{P,m}^Q\}$ is available as side information at π_1. For $k = 2, \ldots, K$, we sequentially identify π_k as follows. We first define

$$\mathcal{I}_k = \mathcal{H} \setminus \bigcup_{i<k} \bigcup_{P:\pi_i \in P} \bigcup_{Q \subset [K]\setminus P} \left\{ w_{P,m}^Q \ : \ m = 1, \ldots, c_P^Q \right\},$$

which is $\mathcal{H}$ minus all the bits demanded by any of $\pi_1, \ldots, \pi_{k-1}$. Thus, any bit in $\mathcal{I}_k$ is demanded by one or more of the nodes in $[K] \setminus \{\pi_1, \ldots, \pi_{k-1}\}$. As $\mathcal{I}_k \subset \mathcal{H}$, it contains an information bit such that this bit is demanded by a node, say $\pi_k \in [K] \setminus \{\pi_1, \ldots, \pi_{k-1}\}$, and the rest of $\mathcal{I}_k$ is not available as side information at π_k.

Observe that $\mathcal{H} = \mathcal{I}_1 \supset \mathcal{I}_2 \supset \cdots \supset \mathcal{I}_K$, and $\mathcal{I}_k \setminus \mathcal{I}_{k+1}$ is the set of bits in $\mathcal{H}$ that are demanded by π_k but not by any of the nodes in $\pi_1, \ldots, \pi_{k-1}$. Thus,

$$\mathcal{I}_1 \setminus \mathcal{I}_2,\ \mathcal{I}_2 \setminus \mathcal{I}_3, \ldots,\ \mathcal{I}_{K-1} \setminus \mathcal{I}_K,\ \mathcal{I}_K$$

form a partition of $\mathcal{H}$. Here, we have abused the notation to denote $\mathcal{I}_K$ by $\mathcal{I}_K \setminus \mathcal{I}_{K+1}$. We also observe that for any choice of k' none of the bits of $\mathcal{I}_{k'}$ is available as side information at $\pi_{k'}$. If $k > k'$, as $\mathcal{I}_k \subset \mathcal{I}_{k'}$, we deduce that none of the bits in $\mathcal{I}_k$ is available as side information at $\pi_{k'}$. Thus, we conclude that each bit in $\mathcal{I}_k \setminus \mathcal{I}_{k+1}$ is demanded by π_k and is neither demanded by and nor available as side information at any of $\pi_1, \ldots, \pi_{k-1}$. Therefore, $|\mathcal{I}_k \setminus \mathcal{I}_{k+1}|$ is upper bounded by the number of bits exclusively demanded by π_k and possibly some subset of $\{\pi_{k+1}, \ldots, \pi_K\}$ and which are also exclusively available at some subset of $\{\pi_{k+1}, \ldots, \pi_K\}$, i.e.,

$$|\mathcal{I}_k \setminus \mathcal{I}_{k+1}| \leq \sum_{\substack{P \subset \{\pi_k, \ldots, \pi_K\} \\ \pi_k \in P}} \sum_{Q \subset \{\pi_{k+1}, \ldots, \pi_K\}} c_P^Q.$$

This provides us the following upper bound,

$$\begin{aligned}\alpha = |\mathcal{H}| &= \sum_{k=1}^{K} |\mathcal{I}_k \setminus \mathcal{I}_{k+1}| \\ &\leq \sum_{k=1}^{K} \sum_{\substack{P \subset \{\pi_k, \ldots, \pi_K\} \\ \pi_k \in P}} \sum_{Q \subset \{\pi_{k+1}, \ldots, \pi_K\}} c_P^Q \\ &\leq \max_{\gamma} \sum_{k=1}^{K} \sum_{\substack{P \subset \{\gamma_k, \ldots, \gamma_K\} \\ \gamma_k \in P}} \sum_{Q \subset \{\gamma_{k+1}, \ldots, \gamma_K\}} c_P^Q,\end{aligned}$$

where the maximization is over all permutations γ of $[K]$.

Lower bound: We derive the lower bound by showing that, for any permutation γ, the set

$$\mathcal{H} = \bigcup_{k=1}^{K} \bigcup_{\substack{P \subset \{\gamma_k, \ldots, \gamma_K\} \\ \gamma_k \in P}} \bigcup_{\substack{Q \subset \\ \{\gamma_{k+1}, \ldots, \gamma_K\}}} \left\{ w_{P,m}^Q \ : \ m = 1, \ldots, c_P^Q \right\}$$

is a generalized independent set. Then,

$$\alpha \geq \max_{\gamma} |\mathcal{H}| = \max_{\gamma} \sum_{k=1}^{K} \sum_{\substack{P \subset \{\gamma_k, \ldots, \gamma_K\} \\ \gamma_k \in P}} \sum_{Q \subset \{\gamma_{k+1}, \ldots, \gamma_K\}} c_P^Q.$$

To show that $\mathcal{H}$ is a generalized independent set, consider any subset $\mathcal{I} \subset \mathcal{H}$. Let k be the smallest integer such that $\mathcal{I}$ contains an information bit $w_{P,m}^Q$ with $\gamma_k \in P$, i.e., k is the smallest integer such that γ_k demands some information bit in $\mathcal{I}$. Therefore, any other bit $w_{P',m'}^{Q'}$ in $\mathcal{I}$ must satisfy

$$P' \subset \{\gamma_{k'}, \ldots, \gamma_K\}, Q' \subset \{\gamma_{k'+1}, \ldots, \gamma_K\} \text{ for some } k' \geq k.$$

Clearly, this bit is not available as side information at γ_k. Thus, $\mathcal{H}$ is a generalized independent set.

Appendix F. Proof of Theorem 6

For unicast problems $c_P^Q > 0$ only if $|P| = 1$. We abuse the notation mildly and use c_k^Q to denote $c_{\{k\}}^Q$.

The Necessity Part: The lower bound of Proposition 3 for unicast problems is $\sum_{i=1}^{K} \sum_{Q \subset \{\gamma_{i+1}, \ldots, \gamma_K\}} c_{\gamma_i}^Q$. For brevity, we will denote this sum as $f(\gamma)$. For Theorem 4 to be tight it is necessary that the bound of this theorem be equal to α, i.e., the value of f be the same for all permutations γ.

We first show that $c_i^{\{j\}} = c_j^{\{i\}}$ for any $i \neq j$. Consider two permutations γ and π which differ only in the two coordinates $K-1, K$, given as, $\gamma_{K-1} = i, \gamma_K = j$ and $\pi_{K-1} = j, \pi_K = i$. Then,

$$\begin{aligned} 0 = f(\gamma) - f(\pi) &= c_i^{\{j\}} + c_i^{\varnothing} + c_j^{\varnothing} - c_j^{\{i\}} - c_j^{\varnothing} - c_i^{\varnothing} \\ &= c_i^{\{j\}} - c_j^{\{i\}}. \end{aligned}$$

This proves the result for $|S| = 2$. Next, we will assume that the necessity part is true for any $S \subset [K]$ of size less than or equal to t, and use induction to prove the result for $|S| = t+1$.

Given any $(t+1)$-set $S \subset [K]$ and any $k, k' \in S$ we will now show that $c_k^{S\setminus k} = c_{k'}^{S\setminus k'}$. Consider two permutations γ, π that differ only in the two coordinates $K-t, K-t+1$, and

$$\begin{aligned} &\gamma_{K-t}, \dots, \gamma_K, \pi_{K-t}, \dots, \pi_K \in S, \\ &\gamma_{K-t} = k, \gamma_{K-t+1} = k', \text{ and } \pi_{K-t} = k', \pi_{K-t+1} = k. \end{aligned}$$

We observe that $S = \{\gamma_{K-t}, \dots, \gamma_K\} = \{\pi_{K-t}, \dots, \pi_K\}$, and

$$\begin{aligned} 0 &= f(\gamma) - f(\pi) \\ &= \sum_{Q \subset \{\gamma_{K-t+1}, \dots, \gamma_K\}} c_k^Q + \sum_{Q \subset \{\gamma_{K-t+2}, \dots, \gamma_K\}} c_{k'}^Q - \sum_{Q \subset \{\pi_{K-t+1}, \dots, \pi_K\}} c_{k'}^Q - \sum_{Q \subset \{\pi_{K-t+2}, \dots, \pi_K\}} c_k^Q \\ &= \sum_{Q \subset S\setminus k} c_k^Q + \sum_{Q \subset S\setminus\{k,k'\}} c_{k'}^Q - \sum_{Q \subset S\setminus k'} c_{k'}^Q - \sum_{Q \subset S\setminus\{k,k'\}} c_k^Q. \end{aligned} \tag{A27}$$

We now argue that except for the two terms $c_k^{S\setminus k}$ and $-c_{k'}^{S\setminus k'}$ all other terms in (A27) cancel out. Consider any term in the first summation of (A27) with $|Q| \leq t-1$. If $k' \in Q$, then by the induction hypothesis, the term $-c_{k'}^{Q\cup k\setminus k'}$ present in the third summation will cancel c_k^Q. If $k' \notin Q$, then $k, k' \notin Q$, and the term $-c_k^Q$ in the fourth summation will cancel c_k^Q. Similarly, every term $c_{k'}^Q$ in the second summation will cancel the corresponding term $-c_{k'}^Q$ in the third summation. It is straightforward to observe that these correspondences between the positive and negative terms are unique, and thus we are left with $0 = c_k^{S\setminus k} - c_{k'}^{S\setminus k'}$.

The Sufficiency Part: The lower bound in Theorem 4 is

$$L^* \geq \sum_{k=1}^{K} \sum_{Q \subset [K]\setminus k} \frac{1}{1+|Q|} c_k^Q = \sum_{k=1}^{K} c_k^{\varnothing} + \sum_{\substack{S \subset [K] \\ |S| \geq 2}} \sum_{k \in S} \frac{c_k^{S\setminus k}}{|S|}.$$

This lower bound can be met by a scheme that uses a combination of uncoded transmission and clique covering. All the bits that are not available at any of the K clients are transmitted uncoded incurring the cost $\sum_{k=1}^{K} c_k^{\varnothing}$. For every $S \subset [K]$ with $|S| \geq 2$, the encoder constructs $|S|$ vectors, one corresponding to each $k \in S$ and broadcasts the XOR of these vectors to the clients. The vector for $k \in S$ consists of the $c_k^{S\setminus k}$ bits demanded by node k and available at nodes $S \setminus k$. All these $|S|$ vectors have the same length $c_k^{S\setminus k}$. These coded transmissions incur an additional cost $\sum_{\substack{S \subset [K] \\ |S| \geq 2}} \sum_{k \in S} \frac{c_k^{S\setminus k}}{|S|}$, thereby achieving the lower bound. This is the well known clique-covering index coding scheme (see [27,28]) and these transmissions allow the clients to decode their demands.

References

1. Maddah-Ali, M.A.; Niesen, U. Fundamental Limits of Caching. *IEEE Trans. Inf. Theory* **2014**, *60*, 2856–2867. [CrossRef]
2. Li, S.; Maddah-Ali, M.A.; Avestimehr, A.S. Coded MapReduce. In Proceedings of the 2015 53rd Annual Allerton Conference on Communication, Control, and Computing (Allerton), Monticello, IL, USA, 29 September–2 October 2015; pp. 964–971. [CrossRef]
3. Li, S.; Maddah-Ali, M.A.; Yu, Q.; Avestimehr, A.S. A Fundamental Tradeoff Between Computation and Communication in Distributed Computing. *IEEE Trans. Inf. Theory* **2018**, *64*, 109–128. [CrossRef]
4. Lee, K.; Lam, M.; Pedarsani, R.; Papailiopoulos, D.; Ramchandran, K. Speeding Up Distributed Machine Learning Using Codes. *IEEE Trans. Inf. Theory* **2018**, *64*, 1514–1529. [CrossRef]
5. Wan, K.; Tuninetti, D.; Ji, M.; Caire, G.; Piantanida, P. Fundamental Limits of Decentralized Data Shuffling. *IEEE Trans. Inf. Theory* **2020**, *66*, 3616–3637. [CrossRef]
6. Elmahdy, A.; Mohajer, S. On the Fundamental Limits of Coded Data Shuffling for Distributed Machine Learning. *IEEE Trans. Inf. Theory* **2020**, *66*, 3098–3131. [CrossRef]
7. Krishnan, P.; Lalitha, V.; Natarajan, L. Coded Data Rebalancing: Fundamental Limits and Constructions. In Proceedings of the 2020 IEEE International Symposium on Information Theory (ISIT), Los Angeles, CA, USA, 21–26 June 2020; pp. 640–645. [CrossRef]
8. El Rouayheb, S.; Sprintson, A.; Sadeghi, P. On coding for cooperative data exchange. In Proceedings of the 2010 IEEE Information Theory Workshop on Information Theory (ITW 2010, Cairo), Cairo, Egypt, 6–8 January 2010; pp. 1–5. [CrossRef]
9. Dau, S.H.; Skachek, V.; Chee, Y.M. Error Correction for Index Coding with Side Information. *IEEE Trans. Inf. Theory* **2013**, *59*, 1517–1531. [CrossRef]
10. Sadeghi, P.; Arbabjolfaei, F.; Kim, Y.H. Distributed index coding. In Proceedings of the 2016 IEEE Information Theory Workshop (ITW), Cambridge, UK, 11–14 September 2016; pp. 330–334. [CrossRef]
11. Li, M.; Ong, L.; Johnson, S.J. Cooperative Multi-Sender Index Coding. *IEEE Trans. Inf. Theory* **2019**, *65*, 1725–1739. [CrossRef]
12. Porter, A.; Wootters, M. Embedded Index Coding. *IEEE Trans. Inf. Theory* **2021**, *67*, 1461–1477. [CrossRef]
13. Yu, Q.; Li, S.; Maddah-Ali, M.A.; Avestimehr, A.S. How to optimally allocate resources for coded distributed computing? In Proceedings of the 2017 IEEE International Conference on Communications (ICC), Paris, France, 21–25 May 2017; pp. 1–7. [CrossRef]
14. Krishnan, P.; Natarajan, L.; Lalitha, V. An Umbrella Converse for Data Exchange: Applied to Caching, Computing, Shuffling & Rebalancing. In Proceedings of the 2020 IEEE Information Theory Workshop, Riva del Garda, Italy, 11–15 April 2020.
15. Wan, K.; Tuninetti, D.; Piantanida, P. On the optimality of uncoded cache placement. In Proceedings of the 2016 IEEE Information Theory Workshop (ITW), Cambridge, UK, 11–14 September 2016; pp. 161–165.
16. Yu, Q.; Maddah-Ali, M.A.; Avestimehr, A.S. The Exact Rate-Memory Tradeoff for Caching with Uncoded Prefetching. *IEEE Trans. Inf. Theory* **2018**, *64*, 1281–1296. [CrossRef]
17. Ji, M.; Caire, G.; Molisch, A.F. Fundamental Limits of Caching in Wireless D2D Networks. *IEEE Trans. Inf. Theory* **2016**, *62*, 849–869. [CrossRef]
18. Asadi, B.; Ong, L.; Johnson, S.J. Centralized Caching with Unequal Cache Sizes. In Proceedings of the 2018 IEEE Information Theory Workshop (ITW), Guangzhou, China, 25–29 November 2018; pp. 1–5.
19. Ibrahim, A.M.; Zewail, A.A.; Yener, A. Centralized Coded Caching with Heterogeneous Cache Sizes. In Proceedings of the 2017 IEEE Wireless Communications and Networking Conference (WCNC), San Francisco, CA, USA, 19–22 March 2017; pp. 1–6.
20. Yapar, C.; Wan, K.; Schaefer, R.F.; Caire, G. On the Optimality of D2D Coded Caching With Uncoded Cache Placement and One-Shot Delivery. *IEEE Trans. Commun.* **2019**, *67*, 8179–8192. [CrossRef]
21. Ibrahim, A.M.; Zewail, A.A.; Yener, A. Device-to-Device Coded Caching with Heterogeneous Cache Sizes. In Proceedings of the 2018 IEEE International Conference on Communications (ICC), Kansas City, MO, USA, 20–24 May 2018; pp. 1–6.
22. Wei, Y.; Ulukus, S. Coded caching with multiple file requests. In Proceedings of the 2017 55th Annual Allerton Conference on Communication, Control, and Computing (Allerton), Monticello, IL, USA, 3–6 October 2017; pp. 437–442.
23. Parrinello, E.; Ünsal, A.; Elia, P. Fundamental Limits of Coded Caching With Multiple Antennas, Shared Caches and Uncoded Prefetching. *IEEE Trans. Inf. Theory* **2020**, *66*, 2252–2268. [CrossRef]
24. Maddah-Ali, M.A.; Niesen, U. Decentralized Coded Caching Attains Order-Optimal Memory-Rate Tradeoff. *IEEE/ACM Trans. Netw.* **2015**, *23*, 1029–1040. [CrossRef]
25. Karat, N.S.; Bhargav, K.L.V.; Rajan, B.S. On the Optimality of Two Decentralized Coded Caching Schemes with and without Error Correction. In Proceedings of the 2020 IEEE International Symposium on Information Theory (ISIT), Los Angeles, CA, USA, 21–26 June 2020; pp. 1664–1669.
26. Bar-Yossef, Z.; Birk, Y.; Jayram, T.S.; Kol, T. Index Coding with Side Information. *IEEE Trans. Inf. Theory* **2011**, *57*, 1479–1494. [CrossRef]
27. Birk, Y.; Kol, T. Informed-source coding-on-demand (ISCOD) over broadcast channels. In Proceedings of the IEEE INFOCOM ’98, the Conference on Computer Communications, Seventeenth Annual Joint Conference of the IEEE Computer and Communications Societies, Gateway to the 21st Century (Cat. No.98), San Francisco, CA, USA, 29 March–2 April 1998; Volume 3, pp. 1257–1264. [CrossRef]
28. Tehrani, A.S.; Dimakis, A.G.; Neely, M.J. Bipartite index coding. In Proceedings of the 2012 IEEE International Symposium on Information Theory Proceedings, Cambridge, MA, USA, 1–6 July 2012; pp. 2246–2250. [CrossRef]