

# Fundamental Limits of Multi-User Distributed Computing of Linearly Separable Functions

K. K. Krishnan Namboodiri, Elizabeth Peter, Derya Malak, and Petros Elia

Communication Systems Department, EURECOM, Sophia-Antipolis, France

E-mail: {karakkad, peter, malak, elia}@eurecom.fr

**Abstract**—This work establishes the fundamental limits of the classical problem of multi-user distributed computing of linearly separable functions. In particular, we consider a distributed computing setting involving  $L$  users, each requesting a linearly separable function over  $K$  basis subfunctions from a master node, who is assisted by  $N$  distributed servers. At the core of this problem lies a fundamental tradeoff between communication and computation: each server can compute up to  $M$  subfunctions, and each server can communicate linear combinations of their locally computed subfunctions’ outputs to at most  $\Delta$  users. The objective is to design a distributed computing scheme that reduces the communication cost (total amount of data from servers to users), and towards this, for any given  $K$ ,  $L$ ,  $M$ , and  $\Delta$ , we propose a distributed computing scheme that jointly designs the task assignment and transmissions, and shows that the scheme achieves optimal performance in the real field under various conditions using a novel converse. We also characterize the performance of the scheme in the finite field using another converse based on counting arguments.

## I. INTRODUCTION

Distributed computing systems are imperative for handling the computationally intensive data-driven tasks that arise in many modern applications. This reality has brought to the fore various distributed computing frameworks, such as MapReduce [1] and Spark [2], that can successfully parallelize computations across clusters of computing nodes. The efficiency of distributed computing is naturally a function — among other things — of the network topology, and of the nature of the requested tasks. A very common setting — which is what we focus on here — considers multiple servers, multiple clients/users, and linearly-separable tasks/functions. In particular, we consider a setting with  $N$  servers and  $L$  users, where each user  $\ell \in [L]$  wishes to compute a distinct function  $F_\ell(\cdot)$  that depends on a set of  $K$  files  $\mathcal{W} = \{W_1, W_2, \dots, W_K\}$  stored at a master node. As one might expect, the functions  $F_\ell(\mathcal{W})$ ,  $\ell \in [L]$ , can be written as the linear combination of  $K$  distinct subfunctions  $f_k(W_k)$ ,  $k \in [K]$ , and the coefficients are captured by a demand matrix  $\mathbf{D}$  of dimension  $L \times K$ . We here assume that the master node coordinates  $N$  servers, and that each server can perform at most  $M \leq K$  subfunction computations, subject to an additional constraint on the maximum number of users that each serve concurrently. We focus

on the practical scenario where each server can communicate messages that are linear functions of their locally computed subfunction outputs, and can do so to a carefully chosen set of up to  $\Delta \leq L$  users. No communication among servers is allowed. In this context, our objective is to design a distributed computing scheme that achieves the minimum communication cost (minimum rate, i.e., minimum amount of information transmitted from servers to users) for a given set of demands under the computational constraint  $M$  and the connectivity constraint  $\Delta$ . The computation–communication tradeoff has been extensively investigated across various distributed computing frameworks [3]–[19].

### A. Related Works

Our considered setting is closest to the distributed computing model in [8], which employs tessellation-based tilings to reduce the number of servers, as well as the work in [9], which considered  $F_\ell(\mathcal{W})$ ’s over a finite field and which designed schemes based on covering codes. Connections can also be found with the work in [5], which addressed a single-user variant of our setting, and which provided bounds derived from matrix-entropic inequalities and covering designs. Naturally, links can also be established with the works in [16], [17] which again study the single-user linearly separable function computation problem in the presence of stragglers, as well as with the work in [22] which considered the distributed computation of a linearly separable Boolean function requested by a single user.

### B. Our Contributions

In this work, we study the multi-user distributed computing problem of linearly separable functions under server–user connectivity and computational constraints. Our contributions are summarized as follows.

- We first propose an achievability scheme (Theorem 1) for the multi-user setting based on a partitioning of the demand matrix into suitable subblocks. For each such subblock, we apply a left nullspace–based construction that jointly designs the dataset assignment and the server transmissions to guarantee exact decodability. This nullspace-based design builds on ideas introduced in our recent work [5], which studied the single-user version of the distributed linearly separable function computation problem. In contrast to [5], which is restricted to a single-user setting, the present work extends the framework

to the multi-user setting, while retaining validity over general system parameters and arbitrary fields.

- We then derive a converse bound on the communication cost that is applicable for computing over finite fields via an equivalent matrix factorization formulation of the problem (Theorem 2). In this formulation, the demand matrix is represented as a product  $\mathbf{D} = \mathbf{C}\mathbf{A}$ , where the sparsity patterns of the factor matrices  $\mathbf{C}$  and  $\mathbf{A}$  are dictated by the  $\Delta$ -connectivity constraint and the  $M$ -computing capability constraint of the servers, respectively. The inner dimension of this factorization corresponds to the communication cost, and the converse is obtained by lower bounding the minimum achievable inner dimension over all such factorizations. Using non-trivial counting arguments, we establish a novel converse on the communication cost. When specialized to the single-user setting considered in [5], the proposed converse is strictly stronger than the previously known bound for every  $q > 2$ . Moreover, under certain divisibility conditions on  $K$ ,  $L$ ,  $M$ , and  $\Delta$ , we show that the achievable communication cost meets the converse as the field size  $q \rightarrow \infty$ . For finite  $q \geq K$ , we show that our scheme is within a factor of 3 from the optimal rate when  $\Delta$  divides  $L$  and  $(\Delta + M - 1)$  divides  $K$ , and within a factor of 8 otherwise (Theorem 3).
- For real fields, we derive an additional converse bound using the same matrix factorization representation of the problem (Theorem 4). In this setting, we count the degrees of freedom involved in factorizing the demand matrix as  $\mathbf{D} = \mathbf{C}\mathbf{A}$  to lower bound the inner dimension, and hence the communication cost. When  $\Delta$  divides  $L$  and  $(\Delta + M - 1)$  divides  $K$ , we show that the achievable rate in Theorem 1 matches this converse bound, and is within a factor of 4 otherwise.

**Notations.** For  $m, n \in \mathbb{Z}^+$ , we let  $[n] = \{1, 2, \dots, n\}$  and  $m|n$  denotes  $m$  divides  $n$ . The number of non-zero entries in a vector  $\mathbf{x}$  is denoted by  $\|\mathbf{x}\|_0$ . A vector  $\mathbf{x}$  is said to be  $k$ -sparse if  $\|\mathbf{x}\|_0 \leq k$ . The cardinality of a set  $\mathcal{S}$  is denoted by  $|\mathcal{S}|$ . For any  $x \in \mathbb{R}^+$ ,  $\lceil x \rceil$  denotes the smallest integer greater than or equal to  $x$ . The vertical concatenation of two matrices  $\mathbf{A}_{m_1 \times n}$  and  $\mathbf{B}_{m_2 \times n}$  are denoted by  $[\mathbf{A}; \mathbf{B}]_{(m_1+m_2) \times n}$ . The  $i$ -th row and the  $j$ -th column of a matrix  $\mathbf{A}_{m \times n}$  are denoted by  $\mathbf{A}(i, :)$  and  $\mathbf{A}(:, j)$ , respectively.

## II. SYSTEM MODEL

Consider a distributed computing system with  $L$  users, each of whom wants to compute a function that depends on a library  $\mathcal{W} = \{W_1, W_2, \dots, W_K\}$  stored at a master node, where  $W_i \in \mathbb{F}^B$  for some field  $\mathbb{F}$  and  $B \in \mathbb{Z}^+$ . The requested functions  $F_\ell(\mathcal{W})$ ,  $\ell \in [L]$ , admit linear separability over  $K$  basis subfunctions  $f_k(W_k)$ ,  $k \in [K]$ , as

$$F_\ell(\mathcal{W}) = \sum_{k=1}^K d_{\ell,k} f_k(W_k), \quad \forall \ell \in [L] \quad (1)$$

where  $d_{\ell,k} \in \mathbb{F}$  and  $f_k(W_k)$  can be a linear or a non-linear function. Let  $\mathbf{D} = [d_{\ell,k}]$ ,  $\ell \in [L]$ ,  $k \in [K]$ , be the matrix

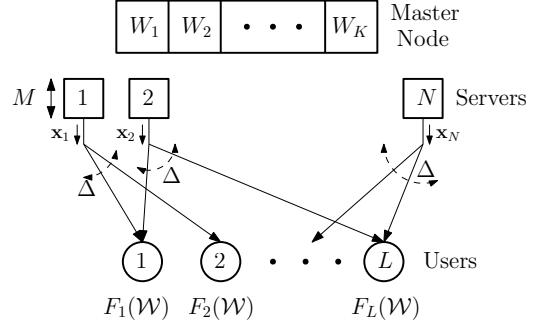


Fig. 1:  $(K, L, M, \Delta)$  multi-user distributed computing system.

formed by the coefficients  $d_{\ell,k}$  in (1). Then, the  $L$  users' demands can be represented as  $\mathbf{F}(\mathcal{W}) = \mathbf{D}\mathbf{f}$ , where  $\mathbf{F}(\mathcal{W}) = [F_1(\mathcal{W}), \dots, F_L(\mathcal{W})]^\top$  and  $\mathbf{f} = [f_1(W_1), \dots, f_K(W_K)]^\top$ . Since  $F_\ell(\mathcal{W})$  is linearly separable, the computation of  $L$  functions is distributed to  $N$  servers with identical computing capabilities. Each server can compute at most  $M \leq K$  subfunctions, and can serve at most  $\Delta \leq L$  users. Figure 1 illustrates the  $L$ -user distributed computing system described above. The system consists of three phases: *demand phase*, *computing phase*, and *communication phase*.

*Demand phase:* In this phase, each user  $\ell \in [L]$  communicates its demand  $F_\ell(\mathcal{W})$  to the master node.

*Computing phase:* Once the demands of all users are known, the master node assigns each server a set of subfunctions to be computed, subject to its computational capacity  $M$  (also referred to as computational cost). Let  $\mathcal{M}_n \subseteq [K]$  denote the indices of the subfunctions computed by server  $n$ , where  $|\mathcal{M}_n| \leq M$ . Then, server  $n$  evaluates all those  $f_k(W_k)$ 's such that  $\mathcal{M}_n \ni k$ . The set  $\mathcal{M} = \{\mathcal{M}_1, \mathcal{M}_2, \dots, \mathcal{M}_N\}$  represents the computational tasks assigned to all  $N$  servers in the system.

*Communication phase:* In this phase, each server linearly encodes its subfunction values and transmits  $r_n$  messages to a subset of users. We consider linear encoding schemes that do not subpacketize the subfunction outputs. The message sent by server  $n$  is denoted by  $x_{n,r}$ ,  $r \in [r_n]$ , and is of the form:  $x_{n,r} = \sum_{k \in \mathcal{M}_n} \alpha_{n,r,k} f_k(W_k)$ ,  $\alpha_{n,r,k} \in \mathbb{F}$ . The message  $\mathbf{x}_n = [x_{n,1}, \dots, x_{n,r_n}]$ ,  $n \in [N]$ , is intended for at most  $\Delta \leq L$  users. Let  $\mathbf{x}$  denote the set of transmissions from all  $N$  servers, and let  $R = \sum_{n \in [N]} r_n$ . Then,  $\mathbf{x}$  can be written as

$$\mathbf{x} = \underbrace{\begin{bmatrix} \alpha_{1,1,1} & \alpha_{1,1,2} & \dots & \alpha_{1,1,K} \\ \vdots & \vdots & \ddots & \vdots \\ \alpha_{N,r_N,1} & \alpha_{N,r_N,2} & \dots & \alpha_{N,r_N,K} \end{bmatrix}}_{\mathbf{A}} \begin{bmatrix} f_1(W_1) \\ \vdots \\ f_K(W_K) \end{bmatrix}$$

where  $\mathbf{A}$  is of dimension  $R \times K$  and  $\|\mathbf{A}(r, :)\|_0 \leq M$ ,  $\forall r \in [R]$ . The communication links between servers and users are assumed to be non-interfering and error-free. The users linearly combine the received messages from different servers to recover the requested functions. The decoding operation at the users can be represented as  $\mathbf{F}(\mathcal{W}) = \mathbf{C}\mathbf{x}$ , where  $\mathbf{C}_{L \times R}$  is composed of the coefficients used to recover the requested

functions from the transmissions. Note that the columns of  $\mathbf{C}$  have a  $\Delta$ -sparsity due to the communication constraint at each server. Therefore,

$$\begin{bmatrix} F_1(\mathcal{W}) \\ \vdots \\ F_L(\mathcal{W}) \end{bmatrix} = \begin{bmatrix} c_{1,1} & c_{1,2} & \dots & c_{1,R} \\ \vdots & \vdots & \ddots & \vdots \\ c_{L,1} & c_{L,2} & \dots & c_{L,R} \end{bmatrix} \mathbf{x}$$

results in the following decomposition

$$\mathbf{D} = \mathbf{C}\mathbf{A} \quad (2)$$

where  $\|\mathbf{C}(:, r)\|_0 \leq \Delta$ ,  $\|\mathbf{A}(r, :)\|_0 \leq M$ ,  $\forall r \in [R]$ . Equation (2) follows from  $\mathbf{F}(\mathcal{W}) = \mathbf{D}\mathbf{f}$  and  $\mathbf{x} = \mathbf{A}\mathbf{f}$ . Hence, the design of a distributed computing scheme can be viewed as the joint design of two matrices  $\mathbf{C}_{L \times R}$  and  $\mathbf{A}_{R \times K}$  with the objective of minimizing  $R$  under  $\Delta$ -sparsity and  $M$ -sparsity constraints on columns of  $\mathbf{C}$  and rows of  $\mathbf{A}$ , respectively.

One of the performance measures of a distributed computing scheme is communication cost, which is defined as the total amount of data transmitted by servers to users during the communication phase, normalized by the size of a message (assuming messages are of unit size). The communication cost required for a given demand matrix  $\mathbf{D}$  under the computational cost  $M$  and per-node serving capacity  $\Delta$  is denoted by  $R_{\mathbf{D}}(M, \Delta)$ . Then,  $R_{\mathbf{D}}(M, \Delta)$  is obtained as  $R_{\mathbf{D}}(M, \Delta) = \sum_{n \in [N]} r_n$ . Our interest is in the worst-case communication cost, denoted by  $R(K, L, M, \Delta)$ , which is defined as

$$R(K, L, M, \Delta) = \max_{\mathbf{D}} R_{\mathbf{D}}(M, \Delta)$$

over all possible  $\mathbf{D}$ 's of dimension  $L \times K$ . The worst-case communication cost is referred to as the rate hereafter. Therefore, the optimal rate, denoted by  $R^*(K, L, M, \Delta)$ , is defined as  $R^*(K, L, M, \Delta) := \inf\{R(K, L, M, \Delta) : R(K, L, M, \Delta) \text{ is achievable}\}$  where the infimum is taken over all possible task assignments and linear transmission and decoding policies satisfying the computational cost  $M$  and the communication constraint  $\Delta$ . We aim to identify the optimal communication-computation costs for a given distributed computing system and design a distributed computing scheme that achieves the optimal rate for a given  $K, L, M$ , and  $\Delta$ .

### III. ACHIEVABILITY RESULTS

In this section, we present an achievable scheme for the multi-user distributed linearly separable function computation problem that applies to all values of  $K, L, M$ , and  $\Delta$ .

**Theorem 1.** *For the  $(K, L, M, \Delta)$  distributed linearly separable function computation problem, the rate*

$$R_{\text{ach}}(K, L, M, \Delta) = \Delta \left\lceil \frac{L}{\Delta} \right\rceil \left\lceil \frac{K}{\Delta + M - 1} \right\rceil \quad (3)$$

is achievable.

*Proof.* Consider  $\mathbf{D} \in \mathbb{F}^{L \times K}$ . We partition  $\mathbf{D}$  into submatrices  $\mathbf{D}_{i,j}$  with  $i \in [\lceil \frac{L}{\Delta} \rceil]$  and  $j \in [\lceil \frac{K}{\Delta + M - 1} \rceil]$  as

$$\mathbf{D} = \begin{bmatrix} \mathbf{D}_{1,1} & \dots & \mathbf{D}_{1, \lceil \frac{K}{\Delta + M - 1} \rceil} \\ \vdots & \ddots & \vdots \\ \mathbf{D}_{\lceil \frac{L}{\Delta} \rceil, 1} & \dots & \mathbf{D}_{\lceil \frac{L}{\Delta} \rceil, \lceil \frac{K}{\Delta + M - 1} \rceil} \end{bmatrix}.$$

Each block  $\mathbf{D}_{i,j}$  has dimension  $\min\{\Delta, L - \Delta(i - 1)\} \times \min\{\Delta + M - 1, K - (\Delta + M - 1)(j - 1)\}$ .

We now design the computing phase and the communication phase for the given demand matrix. First, we consider a sub-demand matrix  $\mathbf{D}_{i,j}$ . Corresponding to  $\mathbf{D}_{i,j}$ , we require  $\Delta$  servers<sup>1</sup>. We refer to them as the servers in group  $(i, j)$ . The set of column indices of  $\mathbf{D}_{i,j}$  is denoted by  $\mathcal{K}_j \triangleq \{(j-1)(\Delta + M - 1) + 1, (j-1)(\Delta + M - 1) + 2, \dots, \min(j(\Delta + M - 1), K)\}$ . Note that  $\mathcal{K}_j$  is also the indices of the files handled by the servers in group  $(i, j)$ . Without loss of generality<sup>2</sup>, we assume that  $\text{rank}(\mathbf{D}_{i,j}) = \Delta$ . Let  $\mathcal{Y}_{i,j} = \{y_1^{(i,j)}, y_2^{(i,j)}, \dots, y_{\Delta}^{(i,j)}\} \subseteq \mathcal{K}_j$  be the indices of the columns that form a  $\Delta \times \Delta$  full-rank submatrix of  $\mathbf{D}_{i,j}$ . Then, we form the task assignment set of the  $\Delta$  servers corresponding to  $\mathbf{D}_{i,j}$  as

$$\mathcal{M}_{i,j} = \{\mathcal{M}_1^{(i,j)}, \mathcal{M}_2^{(i,j)}, \dots, \mathcal{M}_{\Delta}^{(i,j)}\} \quad (4)$$

where, for every  $\delta \in [\Delta]$

$$\mathcal{M}_{\delta}^{(i,j)} = \{y_{\delta}^{(i,j)}\} \cup (\mathcal{K}_j \setminus \mathcal{Y}_{i,j}). \quad (5)$$

That is, the  $\delta$ -th server in the group  $(i, j)$  is assigned to compute the subfunctions  $f_k(W_k)$  for every  $k \in \mathcal{M}_{\delta}^{(i,j)}$ . Each server in group  $(i, j)$  has access to all files indexed by  $\mathcal{K}_j \setminus \mathcal{Y}_{i,j}$ . Moreover, within group  $(i, j)$ , the file indexed by  $y_{\delta}^{(i,j)}$  is uniquely assigned to the  $\delta$ -th server. From (5), it is straightforward to see that  $|\mathcal{M}_{\delta}^{(i,j)}| \leq M$  for every possible  $i, j$ , and  $\delta$ . At this point, note that for a fixed  $i$ , all servers<sup>3</sup> corresponding to  $j \in [\lceil \frac{K}{\Delta + M - 1} \rceil]$  and  $\delta \leq \Delta$  are connected to the users indexed from  $(i - 1)\Delta + 1$  to  $\min(i\Delta, L)$ . This is consistent with the constraint that each server can communicate with at most  $\Delta$  users.

We now design the message transmitted by the  $\delta$ -th server in the group  $(i, j)$ . Note that this message is intended for users indexed from  $(i - 1)\Delta + 1$  to  $\min(i\Delta, L)$ . In order to design the transmission, consider the submatrix of  $\mathbf{D}_{i,j}$ , denoted with  $\mathbf{D}_{i,j}^{\delta}$ , formed by removing the columns indexed by  $\mathcal{M}_{\delta}^{(i,j)}$ . Observe that the size of the matrix  $\mathbf{D}_{i,j}^{\delta}$  is  $\Delta \times (\Delta - 1)$  and  $\text{rank}(\mathbf{D}_{i,j}^{\delta}) = \Delta - 1$ . This is a consequence of the fact that  $\mathbf{D}_{i,j}^{\delta}$  consists of  $\Delta - 1$  linearly independent columns from the originally chosen  $\Delta$  linearly independent columns of  $\mathbf{D}_{i,j}$ . Now, we find a non-trivial vector  $\boldsymbol{\nu}_{i,j}^{\delta} \in \mathbb{F}^{1 \times \Delta}$  in the left nullspace of  $\mathbf{D}_{i,j}^{\delta}$ . Then the transmission from the  $\delta$ -th server in the group  $(i, j)$  is

$$x_{i,j}^{\delta} = \boldsymbol{\nu}_{i,j}^{\delta} \mathbf{D}_{i,j} [f_{(j-1)(\Delta + M - 1) + 1}, \dots, f_{\min(j(\Delta + M - 1), K)}]^{\top}. \quad (6)$$

<sup>1</sup>The total number of servers required is  $N = \Delta \lceil \frac{L}{\Delta} \rceil \lceil \frac{K}{\Delta + M - 1} \rceil$ .

<sup>2</sup>If  $\text{rank}(\mathbf{D}_{i,j}) < \Delta$ —which may occur either due to linearly dependent rows in  $\mathbf{D}_{i,j}$  or because  $\mathbf{D}_{i,j}$  has fewer than  $\Delta$  rows when  $i = \lceil \frac{L}{\Delta} \rceil$ —the rank can be made equal to  $\Delta$  by replacing dependent rows or augmenting the matrix with additional linearly independent rows. Furthermore, the removed rows can be recovered as linear combinations of the remaining rows. If the number of columns itself is less than  $\Delta$ , for  $j = \lceil \frac{K}{\Delta + M - 1} \rceil$ , the case is trivial and will be dealt with later in this proof.

<sup>3</sup>Observe that there is a one-to-one correspondence between servers and task assignments  $\mathcal{M}_{\delta}^{(i,j)}$ .

Note that  $\nu_{i,j}^\delta$  is designed such that  $\nu_{i,j}^\delta \mathbf{D}_{i,j}^\delta = \mathbf{0}$ . Consequently, the support of the vector  $\nu_{i,j}^\delta \mathbf{D}_{i,j}^\delta$  is precisely the set  $\mathcal{M}_\delta^{(i,j)}$ , which enables the  $\delta$ -th server to construct the transmission message in (6). Corresponding to each sub-demand matrix there are  $\Delta$  computing servers, each making a single transmission of subfunction output size<sup>4</sup>. Therefore, the rate achieved by the scheme is

$$R_{\text{ach}}(K, L, M, \Delta) = \Delta \left\lceil \frac{L}{\Delta} \right\rceil \left\lceil \frac{K}{\Delta + M - 1} \right\rceil.$$

Decodability follows from the linear independence of  $\{\nu_{i,j}^\delta : \delta \in [\Delta]\}$  for all  $i, j$ . The detailed proof is given in [23]. This completes the proof of Theorem 1. ■

We now present an example of the proposed scheme and demonstrate the achieved rate.

**Example 1.** Consider the  $(K = 10, L = 6, M = 3, \Delta = 3)$  distributed computing system. Let the demand matrix be

$$\mathbf{D} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 2 & 3 & 4 & 5 & 6 & 2 & 8 & 9 & 10 \\ 1 & 4 & 9 & 5 & 3 & 3 & 5 & 8 & 4 & 1 \\ 1 & 3 & 4 & 9 & 4 & 1 & 2 & 6 & 3 & 10 \\ 1 & 5 & 6 & 3 & 9 & 2 & 4 & 4 & 5 & 1 \\ 1 & 6 & 7 & 1 & 9 & 5 & 10 & 1 & 10 & 10 \end{bmatrix} \in \mathbb{F}_{11}^{6 \times 10}.$$

Upon knowing the user demands, the master node partitions  $\mathbf{D}$  into  $\mathbf{D}_{1,1}$ ,  $\mathbf{D}_{1,2}$ ,  $\mathbf{D}_{2,1}$ , and  $\mathbf{D}_{2,2}$  where each  $\mathbf{D}_{i,j}$  is a matrix of dimension  $3 \times 5$ . Corresponding to each sub-demand matrix  $\mathbf{D}_{i,j}$ , the master node assigns 3 computing servers. As a result, the system requires  $N = 12$  servers. Servers in groups (1, 1) and (1, 2) are connected to users 1, 2, and 3, and servers in groups (2, 1) and (2, 2) are connected to users 3, 4, and 5. Due to space constraints, we describe only the dataset assignment and the transmissions corresponding to the servers in group (1, 1). Note that  $\mathcal{K}_1 = \{1, 2, 3, 4, 5\}$  represents the set of column indices, or equivalently subfunctions, associated with the servers in this group. Further, the first three columns of  $\mathbf{D}_{1,1}$  are linearly independent, and thus we have  $\mathcal{Y}_{1,1} = \{1, 2, 3\}$ . From (5), we have the dataset assignment in the servers in group (1, 1) as follows:

$$\mathcal{M}_1^{(1,1)} = \{1, 4, 5\}, \mathcal{M}_2^{(1,1)} = \{2, 4, 5\}, \mathcal{M}_3^{(1,1)} = \{3, 4, 5\}.$$

Note that the first server does not have access to  $W_2$  and  $W_3$ . Now, we find a vector that resides in the left nullspace of the submatrix  $\mathbf{D}_{1,1}^1$ , which comprises the columns  $\{2, 3\}$  of  $\mathbf{D}_{1,1}$ . Choose a vector, for example,  $\nu_{1,1}^1 = [6, -5, 1]$ , from the left nullspace of  $\mathbf{D}_{1,1}^1$ . Then, the transmission made by the first server is

$$\begin{aligned} x_{1,1}^1 &= [6, -5, 1] \mathbf{D}_{1,1}^1 [f_1(W_1), f_2(W_2), \dots, f_5(W_5)]^\top \\ &= 2f_1(W_1) - 9f_4(W_4) - 16f_5(W_5). \end{aligned} \quad (7)$$

<sup>4</sup>If  $\mathbf{D}_{i,j}^{\lceil \frac{K}{\Delta+M-1} \rceil}$  has fewer than  $\Delta$  columns, the number of computing servers can be chosen to equal the number of columns. Each server computes the subfunction associated with a distinct column index of  $\mathbf{D}_{i,j}^{\lceil \frac{K}{\Delta+M-1} \rceil}$  and transmits the corresponding output to the connected users, yielding a communication cost equal to the number of columns, which is upper bounded by  $\Delta$ .

Similarly, the vectors  $[3, -4, 1]$  and  $[2, -3, 1]$  are in the left nullspaces of  $\mathbf{D}_{1,1}^2$  and  $\mathbf{D}_{1,1}^3$ , respectively. Therefore, the transmissions from server 2 and server 3 in group (1, 1) are

$$\begin{aligned} x_{1,1}^2 &= [3, -4, 1] \mathbf{D}_{1,1}^2 [f_1(W_1), f_2(W_2), \dots, f_5(W_5)]^\top \\ &= -f_2(W_2) - 8f_4(W_4) - 14f_5(W_5) \quad \text{and} \end{aligned} \quad (8)$$

$$\begin{aligned} x_{1,1}^3 &= [2, -3, 1] \mathbf{D}_{1,1}^3 [f_1(W_1), f_2(W_2), \dots, f_5(W_5)]^\top \\ &= 2f_3(W_3) - 5f_4(W_4) - 10f_5(W_5) \end{aligned} \quad (9)$$

respectively. Note that users 1, 2, and 3 have access to  $x_{1,1}^1$ ,  $x_{1,1}^2$ , and  $x_{1,1}^3$ . Therefore, from (7), (8), and (9), the users can decode the following

$$\begin{aligned} \mathbf{D}_{1,1}(1, :) [f_1(W_1), \dots, f_5(W_5)]^\top &= \frac{1}{2}(x_{1,1}^1 - 2x_{1,1}^2 + x_{1,1}^3) \\ \mathbf{D}_{1,1}(2, :) [f_1(W_1), \dots, f_5(W_5)]^\top &= \frac{1}{2}(x_{1,1}^1 - 4x_{1,1}^2 + 3x_{1,1}^3) \\ \mathbf{D}_{1,1}(3, :) [f_1(W_1), \dots, f_5(W_5)]^\top &= \frac{1}{2}(x_{1,1}^1 - 8x_{1,1}^2 + 9x_{1,1}^3). \end{aligned}$$

Similarly, user  $u$ , for  $u = 1, 2, 3$ , can compute  $\mathbf{D}_{1,2}(u, :) [f_6(W_6), \dots, f_{10}(W_{10})]^\top$  from the transmissions sent by the servers in group (1, 2). Finally, user  $u$ , for  $u = 1, 2, 3$ , can compute its requested function as

$$\begin{aligned} F_u(W) &= \mathbf{D}_{1,1}(u, :) [f_1(W_1), \dots, f_5(W_5)]^\top + \\ &\quad \mathbf{D}_{1,2}(u, :) [f_6(W_6), \dots, f_{10}(W_{10})]^\top. \end{aligned}$$

Similarly, users 4, 5, and 6 can decode their requested functions from the transmissions from the servers in groups (2, 1) and (2, 2). Hence, the resulting rate is  $R_{\text{ach}} = 12$ .

For the same parameters  $K = 10$ ,  $L = 6$ , and  $\Delta = 3$ , the scheme presented in [8] for lossless recovery of user demands achieves the same rate  $R = 12$ , but requires a computation cost of  $M = 5$ , whereas our scheme requires only  $M = 3$ . This improvement is enabled by breaking away from the disjoint support assumption adopted in [8]. It is worth noting, however, that our dataset assignment depends on the user demands, while the scheme in [8] allows for a demand-agnostic dataset assignment.

**Remark 1.** The achievability proof builds on the nullspace-based sparsification of the demand matrix proposed in [5], which considers a single-user multiple-function request model and applies the method to sub-blocks of size  $L \times (L + M - 1)$ . In the present multi-user setting, the server-to-user connectivity constraint forces the same approach to be applied to smaller sub-blocks of size  $\Delta \times (\Delta + M - 1)$ .

This construction improves upon the tessellation-based solution in [8], where sub-blocks of size  $\Delta \times M$  are chosen without an additional nullspace-based intra-block sparsification. The improvement, however, requires knowledge of the demand matrix, whereas the scheme in [8] follows a demand-agnostic task assignment strategy.

#### IV. CONVERSE

In this section, we derive a lower bound on the achievable rate of a  $(K, L, M, \Delta)$  distributed computing system. First, we consider the finite field setting where the converse is

developed using some counting arguments in the equivalent matrix factorization formulation. Using the derived lower bound, we demonstrate that our scheme approaches optimal performance as  $q \rightarrow \infty$ . When  $q$  is finite, we show that our scheme is order-optimal. Later, we show that our scheme is optimal over  $\mathbb{R}$  under certain specific conditions by deriving a matching lower bound using the idea of degrees of freedom in the matrix factorization formulation.

**Theorem 2.** *For any  $(K, L, M, \Delta)$  linearly separable distributed computing problem defined over the finite field  $\mathbb{F}_q$ , the optimal rate  $R^*(K, L, M, \Delta)$  follows*

$$R^*(K, L, M, \Delta) \geq \max \left( L, \frac{LK}{\Delta + M + \log_q \binom{L}{\Delta} + \log_q \binom{K}{M} - \log_q(q-1)} \right). \quad (10)$$

*Proof.* Considering the worst-case scenario, we can assume that users' requests are linearly independent in  $f_k(W_k)$ 's. Therefore, we have

$$R^*(K, L, M, \Delta) \geq L. \quad (11)$$

Now, let us look at the  $(K, L, M, \Delta)$  distributed computing problem through the lens of matrix factorization. Given the demand matrix  $\mathbf{D} \in \mathbb{F}_q^{L \times K}$  of rank  $L$ , we need to design two matrices  $\mathbf{C} \in \mathbb{F}_q^{L \times R}$  and  $\mathbf{A} \in \mathbb{F}_q^{R \times K}$ , where  $R \geq L$ , such that  $\mathbf{D} = \mathbf{CA}$  subject to  $\|\mathbf{C}(:, r)\|_0 \leq \Delta$  and  $\|\mathbf{A}(r, :)\|_0 \leq M$ ,  $\forall r \in [R]$ . Given  $L$  and  $K$ , the total possibilities for  $\mathbf{D}$  over a finite field  $\mathbb{F}_q$  is  $q^{LK}$ . Since  $\mathbf{D} = \mathbf{CA}$ , the number of choices for the product  $\mathbf{CA}$  is also  $q^{LK}$ . However, the maximum possible choices for the product  $\mathbf{CA}$  is given by

$$\frac{\left(\binom{L}{\Delta} q^\Delta\right)^R \left(\binom{K}{M} q^M\right)^R}{(q-1)^R}. \quad (12)$$

The numerator in (12) represents the product of the number of distinct  $\mathbf{C}$  and  $\mathbf{A}$  matrices. The matrix  $\mathbf{C}$  is of dimension  $L \times R$  with entries from  $\mathbb{F}_q$ , and its columns are  $\Delta$ -sparse. Therefore,  $\mathbf{C}$  has  $\left(\binom{L}{\Delta} q^\Delta\right)^R$  possibilities. Similarly, the total possibilities for  $\mathbf{A}$  is  $\left(\binom{K}{M} q^M\right)^R$  as  $\mathbf{A} \in \mathbb{F}_q^{R \times K}$  and its rows are  $M$ -sparse. For a chosen  $\mathbf{C}$  and  $\mathbf{A}$ , their scaled versions of the form  $\mathbf{C}' = \mathbf{C}\Lambda$  and  $\mathbf{A}' = \Lambda^{-1}\mathbf{A}$ , where  $\Lambda$  is a  $R \times R$  diagonal matrix defined as  $\Lambda := \text{diag}(\lambda_1, \lambda_2, \dots, \lambda_R)$ , also result in the same product. Note that the columns of  $\mathbf{C}'$  and rows of  $\mathbf{A}'$  are  $\Delta$ -sparse and  $M$ -sparse, respectively, as  $\lambda_r \neq 0$  for all  $r \in [R]$ . Notice that the same product matrix is obtained if any other dense transformation matrix is used in place of  $\Lambda$ . However, the resulting  $\mathbf{C}'$  and  $\mathbf{A}'$  do not satisfy the required sparsity constraints. Therefore, the number of distinct product matrices  $\mathbf{C}$  and  $\mathbf{A}$  jointly generate is given by (12). Thus, we get the following relation

$$q^{LK} \leq \frac{\left(\binom{L}{\Delta} q^\Delta\right)^R \left(\binom{K}{M} q^M\right)^R}{(q-1)^R}. \quad (13)$$

Taking the logarithm of (13) and rearranging gives

$$R \geq \frac{LK}{\Delta + \log_q \binom{L}{\Delta} + M + \log_q \binom{K}{M} - \log_q(q-1)}. \quad (14)$$

Since  $R^*(K, L, M, \Delta)$  is given by the infimum of achievable rates, we obtain (10) from (11) and (14). ■

**Corollary 1.** *When the field size  $q \rightarrow \infty$ , the lower bound on the optimal rate becomes  $R^*(K, L, M, \Delta) \geq \frac{LK}{\Delta + M - 1}$ .*

*Proof.* The proof follows directly from (10). As  $q \rightarrow \infty$ , the terms  $\log_q \binom{L}{\Delta} \rightarrow 0$ ,  $\log_q \binom{K}{M} \rightarrow 0$ , and  $\log_q(q-1) \rightarrow 1$ . ■

**Remark 2.** *Corollary 1 implies that as the field size  $q \rightarrow \infty$ , the lower bound in (14) matches the achievable rate in Theorem 1 when  $\Delta|L$  and  $(\Delta + M - 1)|K$ .*

**Theorem 3** (Optimality gap). *When  $\Delta|L$ ,  $(\Delta + M - 1)|K$  and  $q \geq K$ , the achievable rate in Theorem 1 satisfies*

$$1 \leq \frac{R_{\text{ach}}(K, L, M, \Delta)}{R^*(K, L, M, \Delta)} \leq 3.$$

*Proof.* The proof follows by bounding the ratio of  $R_{\text{ach}}$  and  $R^*$  in (3) and (10), respectively, using  $\binom{n}{k} \leq n^k$  and approximating the logarithmic terms under the assumption  $q \geq K$ . The details are available in [23]. ■

Note that when conditions  $\Delta|L$  and  $(\Delta + M - 1)|K$  do not hold, the rate is within a factor of 8 from the lower bound. When the underlying field is  $\mathbb{R}$ , we derive a lower bound on the rate similar to Theorem 2 using the degrees of freedom of  $\mathbf{D}$  and the degrees of freedom of  $\mathbf{C}$  and  $\mathbf{A}$ .

**Theorem 4.** *For a  $(K, L, M, \Delta)$  linearly separable distributed computing problem defined over  $\mathbb{R}$ , the optimal rate satisfies*

$$R^*(K, L, M, \Delta) \geq \max \left( L, \frac{LK}{\Delta + M - 1} \right). \quad (15)$$

*Proof.* A demand matrix  $\mathbf{D} \in \mathbb{R}^{L \times K}$  of rank  $L$  has  $LK$  degrees of freedom (DoF).<sup>5</sup> Since  $\mathbf{D} = \mathbf{CA}$ , we have  $\text{DoF}(\mathbf{D}) = \text{DoF}(\mathbf{CA}) \leq \text{DoF}(\mathbf{C}) + \text{DoF}(\mathbf{A})$ . The matrix  $\mathbf{C} \in \mathbb{R}^{L \times R}$  has  $\Delta$ -sparse columns, therefore,  $\text{DoF}(\mathbf{C}) = \Delta R$ . Similarly,  $\text{DoF}(\mathbf{A}) = MR$  as the matrix  $\mathbf{A} \in \mathbb{R}^{R \times K}$  has  $M$ -sparse rows. Thus, we get  $LK \leq \Delta R + MR$ , which can be tightened by considering the redundant DoF as done in the finite field case (Theorem 2). i.e., the scaled versions of  $\mathbf{C}$  and  $\mathbf{A}$  having the following form  $\mathbf{C}\Lambda$  and  $\Lambda^{-1}\mathbf{A}$ , where  $\Lambda \in \mathbb{R}^{R \times R}$  is a diagonal matrix of rank  $R$ , also result in the same product. Therefore, we get  $\text{DoF}(\mathbf{D}) \leq \text{DoF}(\mathbf{C}) + \text{DoF}(\mathbf{A}) - \text{DoF}(\Lambda)$ . Notice that the  $\text{DoF}(\Lambda) = R$  as  $\Lambda \in \mathbb{R}^{R \times R}$  is a diagonal matrix of rank  $R$ . Thus, we have  $LK \leq \Delta R + MR - R$  which results in  $R \geq \frac{LK}{\Delta + M - 1}$ . This completes the proof of Theorem 4. ■

**Remark 3.** *For the real field, the achievable rate in Theorem 1 is optimal when  $\Delta|L$  and  $(\Delta + M - 1)|K$ ; otherwise, it is within a factor of 4 from the optimal rate.*

<sup>5</sup>The degrees of freedom of a matrix is defined in [24], [25]. The degrees of freedom of a matrix can be viewed as the number of independent parameters that can be varied in the matrix.

## REFERENCES

- [1] J. Dean and S. Ghemawat, "MapReduce: Simplified data processing on large clusters," *Commun. ACM*, vol. 51, no. 1, pp. 107–113, Jan. 2008.
- [2] M. Zaharia, M. Chowdhury, M. J. Franklin, S. Shenker, and I. Stoica, "Spark: Cluster computing with working sets," in *USENIX Workshop Hot Topics in Cloud Comput. (HotCloud 10)*, Jun. 2010.
- [3] S. Li, M. A. Maddah-Ali, Q. Yu, and A. S. Avestimehr, "A fundamental tradeoff between computation and communication in distributed computing," *IEEE Trans. Inf. Theory*, vol. 64, no. 1, pp. 109–128, 2018.
- [4] Q. Yan, S. Yang, and M. Wigger, "Storage-computation-communication tradeoff in distributed computing: Fundamental limits and complexity," *IEEE Trans. Inf. Theory*, vol. 68, no. 8, pp. 5496–5512, 2022.
- [5] K. K. K. Namboodiri, E. Peter, D. Malak, and P. Elia, "Fundamental limits of distributed computing for linearly separable functions," *arXiv:2509.23447*, Sep. 2025.
- [6] M. Ye and E. Abbe, "Communication-computation efficient gradient coding," in *Proc. Int. Conf. Mach. Learn. (ICML)*, 2018, pp. 5610–5619.
- [7] R. Tandon, Q. Lei, A. G. Dimakis, and N. Karampatziakis, "Gradient coding: Avoiding stragglers in distributed learning," in *Proc. Int. Conf. Mach. Learn. (ICML)*, 2017, pp. 3368–3376.
- [8] A. Khalesi and P. Elia, "Tessellated distributed computing," *IEEE Trans. Inf. Theory*, vol. 71, no. 6, pp. 4754–4784, Apr. 2025.
- [9] —, "Multi-user linearly-separable distributed computing," *IEEE Trans. Inf. Theory*, vol. 69, no. 10, pp. 6314–6339, Jun. 2023.
- [10] D. Malak, M. R. Deylam Salehi, B. Serbetci, and P. Elia, "Multi-server multi-function distributed computation," *Entropy*, vol. 26, no. 6, 2024.
- [11] F. Brunero and P. Elia, "Multi-access distributed computing," *IEEE Trans. Inf. Theory*, vol. 70, no. 5, pp. 3385–3398, 2024.
- [12] E. Peter, K. K. K. Namboodiri, and B. S. Rajan, "Wireless MapReduce arrays for coded distributed computing," in *Proc. IEEE Inf. Theory Workshop (ITW)*, 2024, pp. 163–168.
- [13] E. Parrinello, E. Lampir, and P. Elia, "Coded distributed computing with node cooperation substantially increases speedup factors," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, 2018, pp. 1291–1295.
- [14] Q. Yu, M. A. Maddah-Ali, and A. S. Avestimehr, "Straggler mitigation in distributed matrix multiplication: Fundamental limits and optimal coding," *IEEE Trans. Inf. Theory*, vol. 66, no. 3, pp. 1920–1933, 2020.
- [15] Q. Yu, S. Li, N. Raviv, S. M. M. Kalan, M. Soltanolkotabi, and S. A. Avestimehr, "Lagrange coded computing: Optimal design for resiliency, security, and privacy," in *Proc. Int. Conf. Artif. Intell. Statist. (AISTATS)*, 2019, pp. 1215–1225.
- [16] K. Wan, H. Sun, M. Ji, and G. Caire, "Distributed linearly separable computation," *IEEE Trans. Inf. Theory*, vol. 68, no. 2, pp. 1259–1278, Nov. 2021.
- [17] —, "On the tradeoff between computation and communication costs for distributed linearly separable computation," *IEEE Trans. Commun.*, vol. 69, no. 11, pp. 7390–7405, Aug. 2021.
- [18] K. Lee, M. Lam, R. Pedarsani, D. Papailiopoulos, and K. Ramchandran, "Speeding up distributed machine learning using codes," *IEEE Trans. Inf. Theory*, vol. 64, no. 3, pp. 1514–1529, 2018.
- [19] S. Dutta, M. Fahim, F. Haddadpour, H. Jeong, V. Cadambe, and P. Grover, "On the optimal recovery threshold of coded matrix multiplication," *IEEE Trans. Inf. Theory*, vol. 66, no. 1, pp. 278–301, 2020.
- [20] J. Maheri and P. Elia, "Constructing hamiltonian decompositions of complete K-uniform hypergraphs," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, 2025, pp. 1–6.
- [21] A. Khalesi and P. Elia, "Hyper-minrank: A unified hypergraph characterization of multi-sender index coding," *arXiv preprint 2512.13615*, 2025.
- [22] A. Tanha and D. Malak, "The influence of placement on transmission in distributed computing of boolean functions," in *Proc. IEEE 25th Int. Workshop Signal Process. Adv. Wireless Commun. (SPAWC)*, Sep. 2024, pp. 816–820.
- [23] K. K. K. Namboodiri, E. Peter, D. Malak, and P. Elia, "Fundamental limits of multi-user distributed computing of linearly separable functions," *arXiv preprint 2601.10603*, 2026.
- [24] H. C. Andrews and K. L. Caspari, "Degrees of freedom and modular structure in matrix multiplication," *IEEE Trans. Computers*, vol. C-20, pp. 133–141, 1971.
- [25] A. Uschmajew and B. Vandereycken, "Geometric methods on low-rank matrix and tensor manifolds," in *Handbook of variational methods for nonlinear geometric data*. Springer, 2020, pp. 261–313.