New Achievability Schemes for Distributed Computing of Linearly Separable Functions

Elizabath Peter, K. K. Krishnan Namboodiri, Derya Malak, and Petros Elia
Communication Systems Department, EURECOM, Biot, France
Email: {peter, karakkad, malak, elia}@eurecom.fr

***Abstract*—This work addresses the classical problem of distributed computation of linearly separable functions, where a master node with access to K datasets employs N servers to compute L user-requested functions, each defined over the datasets. Servers are assigned subfunctions of the datasets and transmit computed outputs to the user, who reconstructs the demanded outputs. The central challenge is to minimize both the per-server computational load and the communication cost from servers to the user, while ensuring recovery for any possible set of L demands. For any given K, L, and M, we propose a distributed computing scheme that achieves the optimal communication cost when $K < L + M$. When $M \geq K/2$, we present an alternative scheme that yields a lower communication cost than the former. The key innovation in both schemes is a nullspace-based design principle that governs both dataset assignment and server transmissions, ensuring exact decodability for all demands.**

I. Introduction

Recent advances in machine learning have made distributed computing essential for large-scale workloads. Frameworks such as MapReduce [1] and Spark [2] partition tasks across multiple servers, overcoming the limitations of individual nodes. These systems, however, face challenges from stragglers [3]–[7], data privacy, and adversarial behavior [8], [9], while being fundamentally constrained by computational capability and communication bandwidth. These intertwined constraints give rise to the *communication–computation tradeoff*, which underpins the design of distributed computing frameworks including MapReduce [10]–[15], gradient coding/ distributed function computing [3], [16], [17], and distributed matrix multiplication [5], [6], [18].

This fundamental tradeoff lies at the core of our study, where we examine the (K, L, M) distributed linearly separable function computation problem over a field $\mathbb{F}$. In this setting, a user requests L function values from a master node, each corresponding to the output of an independent function $F_\ell,\ \ell \in [L]$, acting on a dataset library $\mathcal{W}$ of K independent and identically distributed (i.i.d.) datasets. These functions are linearly separable over K basis subfunctions $f_j,\ j \in [K]$, which enables the master node to parallelize computation across N distributed servers. Each server, with limited computing power, is assigned a set of at most M subfunctions to compute, each evaluated on an i.i.d. dataset stored locally. After computation, servers transmit linear combinations of their computed values to the user, who then recovers the L requested outputs. The objective, for any (K, L, M) instance, is to design a distributed computing scheme that jointly allocates subfunctions to servers and specifies transmissions so as to minimize the total communication cost while ensuring exact recovery of all requested functions.

In the extremes, $M = K$ corresponds to the centralized setting where a single server transmits L messages, while $M = 1$ requires $N = K$ servers and incurs a communication cost of K. The intermediate regime $1 < M < K$ presents the main combinatorial challenge, where one must jointly allocate subfunctions across servers and design transmissions so as to minimize the total communication cost.

A. Related Works

Our work is most closely related to [19], [20], which study a similar problem of linearly separable function computation with N servers, a single user, multiple requested functions over K datasets, and an objective of minimizing communication cost for a given M. The main distinction with our work is that [19], [20] emphasize on straggler mitigation, and on the specific case of the cyclic task assignment. Related work can also be found in the more recent [21], [22], which consider a multi-user distributed computing problem, where each server is connected to multiple, but not all, users, each asking for their own desired function. Both these works [21], [22] aim to reduce the communication and computation costs, the first [21] by exploiting the properties of covering codes, and the second [22] by employing optimal tilings from tessellation theory. The multi-user distributed computing scheme in [22] – once translated onto our single-user setting – operates under various restrictive assumptions of disjoint tasks across servers, which do not appear in our work.

B. Contributions

In this work, we study the distributed linearly separable function computation problem with the goal of minimizing, for a given computation cost M, the total communication cost R from servers to the user. Since each server can compute at most $M \leq K$ subfunctions, the key challenge is to jointly design (i) task assignments that respect the per-server budget and (ii) transmissions that guarantee exact recovery of all requested functions. Performance is measured by the worst-case communication cost across all possible demands.

We introduce a novel technique that provides a sufficient condition for designing both task assignments and server transmissions. The core idea (Lemma 1) leverages the existence of left nullspaces of carefully selected submatrices of the demand matrix to guarantee exact decodability when $K \leq L + M - 1$. In this regime, a 'nullspace-based matrix' (cf. (4)) can test feasibility for any task assignment and, if feasible, directly specify the transmissions. Moreover, for all values of K, L, M satisfying $K \leq L + M - 1$, we explicitly design feasible assignments and transmission schemes achieving the globally optimal communication cost. While prior works [19], [20] employed nullspace ideas solely for transmission design, our approach extends this principle to task assignment itself. The key distinction is that the nullspace design dictates the task allocation, unlike [19], [20], where cyclic assignment predetermined the nullspace structure. This extension ensures exact decodability for all demands, going beyond the probabilistic guarantees in [19], [20].[1]

A few additional design elements appear in the subsequent Schemes 1 and 2, designed for the case of $K > L + M - 1$.

- Scheme 1 (Theorem 1) applies a column-wise partition of the demand matrix into submatrices, on each of which the nullspace-based design for task assignment and transmissions is applied independently. The scheme works for any (K, L, M), and the corresponding communication cost takes the form $R = \min(K, L\lceil K/(L + M - 1)\rceil)$.
- Scheme 2 (Theorem 2) is distinguished by admitting a task assignment independent of the demand matrix. It applies in the regime $M \geq K/2$, where we augment the demand matrix with carefully designed virtual demand, enabling a more efficient fusion of the partitioning and nullspace approaches.

C. Notations

For a positive integer n, we let $[n] = \{1, 2, \ldots, n\}$. For m, $n \in \mathbb{Z}^+$ such that $m < n$, $[m : n]$ denotes the set $\{m, m+1, \ldots, n\}$, and $m \mid n$ denotes m divides n. All vectors are assumed to be column vectors. For a vector $\mathbf{x}$, the number of non-zero entries in $\mathbf{x}$ is denoted by $||\mathbf{x}||_0$. For any set $\mathcal{S}$, the cardinality of $\mathcal{S}$ is denoted by $|\mathcal{S}|$. The complement of set $\mathcal{S}$ is denoted by $\mathcal{S}^c$. The vertical concatenation of two matrices $\mathbf{A}_{m_1\times n}$ and $\mathbf{B}_{m_2\times n}$ are denoted by $[\mathbf{A}; \mathbf{B}]$. 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. For a set $\mathcal{S} \subseteq [n]$ and a matrix $\mathbf{A}$ with n columns, $\mathbf{A}_{\mathcal{S}}$ denotes the submatrix of $\mathbf{A}$ formed by the columns indexed by $\mathcal{S}$. For any $x \in \mathbb{R}^+$, $\lceil x \rceil$ denotes the smallest positive integer greater than or equal to x, and $\lfloor x \rfloor$ denotes the largest positive integer less than or equal to x. A vector of length n with all zeros is denoted by $\mathbf{0}_{n\times 1}$. An $n-$length unit vector with a one at the i-th position and zeroes elsewhere is denoted by $\mathbf{e}_i$. The support of a vector $\mathbf{x} \in \mathbb{F}^n$ is defined as $supp(\mathbf{x}) := \{i \in [n] : x_i \neq 0\}$.

II. System Model and Problem Formulation

Consider a distributed computing system with a master node having a library of K i.i.d. datasets $\mathcal{W} = \{W_1, W_2, \ldots, W_K\}$, where $W_k \in \mathbb{F}^B$, $\forall k \in [K]$. In this setting, a single user wishes to compute $L \leq K$ independent functions $F_1, F_2, \ldots, F_L$ on $\mathcal{W}$, where $F_\ell : (\mathbb{F}^B)^K \to \mathbb{F}^T$, $\forall \ell \in [L]$, and T denotes the size of each function value (a vector of T elements from the field). Furthermore, each function $F_\ell, \forall \ell \in [L]$, is linearly decomposable as

$$F_\ell(\mathcal{W}) = d_{\ell,1}f_1(W_1) + d_{\ell,2}f_2(W_2) + \cdots + d_{\ell,K}f_K(W_K),$$

where $d_{\ell,j} \in \mathbb{F}$, and the subfunction $f_j : \mathbb{F}^B \to \mathbb{F}^T$, $j \in [K]$, is assumed to be bounded and can be linear or non-linear and can be computationally intensive. Consequently, the L requested functions can be represented as

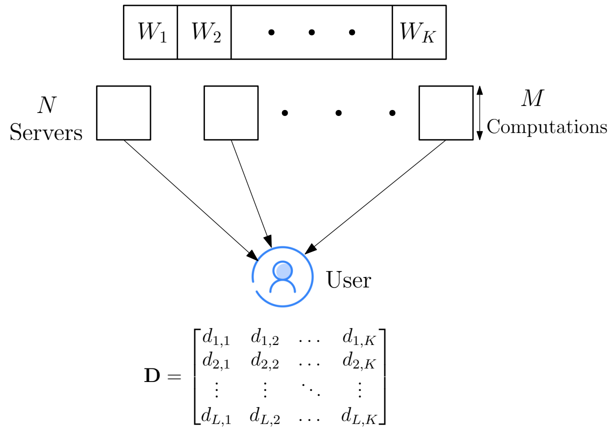

Fig. 1: System model

$$\begin{bmatrix} F_1(\mathcal{W}) \\ F_2(\mathcal{W}) \\ \vdots \\ F_L(\mathcal{W}) \end{bmatrix} = \underbrace{\begin{bmatrix} d_{1,1} & d_{1,2} & \ldots & d_{1,K} \\ d_{2,1} & d_{2,2} & \ldots & d_{2,K} \\ \vdots & \vdots & \ddots & \vdots \\ d_{L,1} & d_{L,2} & \ldots & d_{L,K} \end{bmatrix}}_{\mathbf{D}} \begin{bmatrix} f_1(W_1) \\ f_2(W_2) \\ \vdots \\ f_K(W_K) \end{bmatrix} \quad (1)$$

where – under the worst-case assumption – we consider the case of $\text{rank}(\mathbf{D}) = L$. Since the L functions are linearly separable, the computation of the $F_\ell(\mathcal{W})$'s can be performed in a distributed manner across the N servers. We consider the case where the computational capability of the servers is limited, in that each server can compute a maximum of $M \leq K$ subfunctions. This model is illustrated in Figure 1.

The system operates in three phases: *demand phase*, *computing phase*, and *communication phase*.

Demand phase: In the demand phase, the user informs the master node of its L desired functions, which are naturally described by the matrix $\mathbf{D} \in \mathbb{F}^{L\times K}$.

Computing phase: Subject to the constraint that each server computes at most M subfunctions, the master uses the demand matrix $\mathbf{D}$ to assign to server n a subset $\mathcal{M}_n \subset [K]$ of subfunctions and the associated datasets. At the end of this phase, server n evaluates $f_j(W_j) \in \mathbb{F}^T$ for all $j \in \mathcal{M}_n$. We often refer to the collection $\mathcal{M} = \{\mathcal{M}_n : n \in [N]\}$ as the *task assignment*.

Communication phase: After computing, server n sends r_n linearly encoded messages based on its local outputs $f_j(W_j)$, where each message is of the form[2]

$$x_{n,r} = \sum_{j\in\mathcal{M}_n} \alpha_{n,r,j} f_j(W_j) \quad (2)$$

for $r \in [r_n]$, with coefficients $\alpha_{n,r,j} \in \mathbb{F}$. This means that

[1] Indeed, for small finite field sizes, those schemes may fail on a significant fraction of demand matrices (see Table I in [20]).

[2] We adopt the standard no-subpacketization assumption, where servers do not divide subfunction outputs into smaller parts.

each server n transmits $\mathbf{x}_n = [x_{n,1}, x_{n,2}, \ldots, x_{n,r_n}]^\intercal$, where

$$\begin{bmatrix} x_{n,1} \\ x_{n,2} \\ \vdots \\ x_{n,r_n} \end{bmatrix} = \underbrace{\begin{bmatrix} \alpha_{n,1,1} & \alpha_{n,1,2} & \ldots & \alpha_{n,1,K} \\ \alpha_{n,2,1} & \alpha_{n,2,2} & \ldots & \alpha_{n,2,K} \\ \vdots & \vdots & \ddots & \vdots \\ \alpha_{n,r_n,1} & \alpha_{n,r_n,2} & \ldots & \alpha_{n,r_n,K} \end{bmatrix}}_{\mathbf{A}_n \in \mathbb{F}^{r_n \times K}} \begin{bmatrix} f_1(W_1) \\ f_2(W_2) \\ \vdots \\ f_K(W_K) \end{bmatrix}.$$

Since $x_{n,r}$ is a linear combination of at most M subfunctions, then $||\mathbf{A}_n(r,:)||_0 \leq M, \forall r \in [r_n]$ and $n \in [N]$. For $\mathbf{x} = [\mathbf{x}_1; \mathbf{x}_2; \ldots; \mathbf{x}_N]$ denoting the entire set of transmissions across all the servers, for $R = \sum_{n\in[N]} r_n$, and for $\mathbf{A} = [\mathbf{A}_1; \mathbf{A}_2; \ldots; \mathbf{A}_N] \in \mathbb{F}^{R\times K}$ denoting the *encoding matrix*, the entire set of transmissions can be represented as

$$\mathbf{x} = \mathbf{A}[f_1(W_1), f_2(W_2), \ldots, f_K(W_K)]^\intercal$$

where each row of $\mathbf{A}$ has a non-empty support of cardinality at most M.

The communication link between each of the N servers and the user is assumed to be error-free and non-interfering. The user is allowed to perform linear operations on the received messages, in order to decode the L function values.

The communication cost here represents the total number of transmissions, from the servers to the user, required for recovery of the L desired function outputs at the user. In particular, for a given $\mathbf{D} \in \mathbb{F}^{L\times K}$, the communication cost is defined as $R_{\mathbf{D}}(M) = \sum_{n\in[N]} r_n$, where we assume unit-length file messages, i.e., we assume $|x_{n,r}| = 1, \forall n \in [N], r \in [r_n]$.[3] The rate of interest in our work will represent the worst-case communication cost

$$R(K, L, M) = \max_{\mathbf{D}\in\mathbb{F}^{L\times K}} R_{\mathbf{D}}(M)$$

over all full-rank matrices $\mathbf{D}$, and thus our interest is in identifying the optimal rate

$$R^*(K, L, M) = \inf\{R(K, L, M) \colon R(K, L, M) \text{ is achievable}\}$$

where the infimum is over all task assignments and linear transmission policies. Our objective is to design the assignment and transmission scheme that approaches this optimum.

III. Main Results

In this section, we present two achievable schemes for the distributed linearly separable function computation problem. The first scheme (Theorem 1) applies to all values of (K, L, M), while the second scheme (Theorem 2) is designed to improve the performance of the first scheme in the region of $M \geq K/2$. Additionally, the second scheme enjoys a demand-agnostic task assignment.

A. Scheme 1: A General Achievability Scheme

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

$$R_1(K, L, M) = \min\left\{K, L\left\lceil\frac{K}{L+M-1}\right\rceil\right\} \tag{3}$$

is achievable.

[3] A subfunction output/coded transmission vector of length T is treated as one unit.

Proof. The proof of the achievability of the rate expression in (3) is divided into two cases. Case 1 for $K \leq L+M-1$, and Case 2 for $K > L+M-1$. The crux of the proof of Case 1 lies in Lemma 1 presented next, while for Case 2, we combine the idea presented in Lemma 1 with a demand matrix partitioning technique.

*1) **Case 1** $(K \leq L+M-1)$:* We first consider the case $K \leq L+M-1$. It is sufficient to focus on the scenario $K = L+M-1$, because if $K < L+M-1$, each server uses only $M' = K-L+1 < M$ of its computing capability, reducing the problem to the case $K = L+M'-1$. Consider a demand matrix $\mathbf{D} \in \mathbb{F}^{L\times K}$, where $\mathbf{D} = [\mathbf{d}_1, \mathbf{d}_2, \ldots, \mathbf{d}_K]$. Recall that $\mathcal{M}_n \subset [K]$ is the set of indices of the datasets known to server n, where $|\mathcal{M}_n| = M = K-L+1, \forall n \in [N]$. For every $n \in [N]$, we now define a submatrix $\mathbf{D}_{\mathcal{M}_n} = [\mathbf{d}_j : j \in \mathcal{M}_n]$. Consider the following matrix

$$\mathbf{N}_{\mathbf{D},\mathcal{M}} = [\mathcal{N}(\mathbf{D}^\intercal_{\mathcal{M}_1^c}), \mathcal{N}(\mathbf{D}^\intercal_{\mathcal{M}_2^c}), \ldots, \mathcal{N}(\mathbf{D}^\intercal_{\mathcal{M}_N^c})]^\intercal \tag{4}$$

where $\mathcal{N}(.)$ is the nullspace operator, which outputs a set of basis vectors of the nullspace of the matrix on which the operator acts. For any $\mathcal{M}_n$ with $|\mathcal{M}_n| = M = K-L+1$, the submatrix $\mathbf{D}_{\mathcal{M}_n^c}$ is of size $L \times (L-1)$, and therefore $\mathbf{D}_{\mathcal{M}_n^c}$ has a non-trivial left nullspace. Consequently, the rank of the left nullspace of $\mathbf{D}_{\mathcal{M}_n^c}$ is at least one, and $\mathcal{N}(\mathbf{D}^\intercal_{\mathcal{M}_n^c})$ contains at least one vector for every $n \in [N]$. Thus, the number of rows in the matrix $\mathbf{N}_{\mathbf{D},\mathcal{M}}$ is at least N. We now have the following lemma.

Lemma 1. *For a given $\mathbf{D} \in \mathbb{F}^{L\times K}$ with rank$(\mathbf{D}) = L$, and $\mathcal{M} = \{\mathcal{M}_1, \mathcal{M}_2, \ldots, \mathcal{M}_N\}$, the rate L is achievable if*

$$\text{rank}(\mathbf{N}_{\mathbf{D},\mathcal{M}}) = L. \tag{5}$$

Furthermore, for any given $K, L,$ and M with $K = L+M-1$, and for any $\mathbf{D} \in \mathbb{F}^{L\times K}$, the optimal rate

$$R^*(K, L, M) = L$$

is achievable with $N = L$ servers.

Proof of Lemma 1. Consider a demand matrix $\mathbf{D} \in \mathbb{F}^{L\times K}$, where $\text{rank}(\mathbf{D}) = L$. Furthermore, assume that the task assignment $\mathcal{M} = \{\mathcal{M}_1, \mathcal{M}_2, \ldots, \mathcal{M}_N\}$ is such that the condition $\text{rank}(\mathbf{N}_{\mathbf{D},\mathcal{M}}) = L$ holds. Therefore, the matrix $\mathbf{N}_{\mathbf{D},\mathcal{M}}$ has at least a set of L independent rows. Let $\tilde{\mathbf{N}}_{\mathbf{D},\mathcal{M}}$ be an $L \times L$ matrix constructed by choosing a set of L independent rows from $\mathbf{N}_{\mathbf{D},\mathcal{M}}$. In the communication phase, there is a transmission corresponding to each row in $\tilde{\mathbf{N}}_{\mathbf{D},\mathcal{M}}$. Let the ℓ-th row of $\tilde{\mathbf{N}}_{\mathbf{D},\mathcal{M}}$, denoted by $\tilde{\mathbf{N}}_{\mathbf{D},\mathcal{M}}(\ell,:)$, correspond to the task assignment $\mathcal{M}_n \in \mathcal{M}$ for some $n \in [N]$. Then, server n makes the following transmission

$$\tilde{\mathbf{N}}_{\mathbf{D},\mathcal{M}}(\ell,:)\mathbf{D}[f_1(W_1), f_2(W_2), \ldots, f_K(W_K)]^\intercal. \tag{6}$$

First, we show that server n can construct the transmission in (6) from the available datasets in $\mathcal{M}_n$. In other words, we show that

$$supp(\tilde{\mathbf{N}}_{\mathbf{D},\mathcal{M}}(\ell,:)\mathbf{D}) \subseteq \mathcal{M}_n. \tag{7}$$

However, notice that $\tilde{\mathbf{N}}_{\mathbf{D},\mathcal{M}}(\ell,:) \in \mathcal{N}(\mathbf{D}^\intercal_{\mathcal{M}_n^c})$ and, therefore, we have $\tilde{\mathbf{N}}_{\mathbf{D},\mathcal{M}}(\ell,:)\mathbf{D}_{\mathcal{M}_n^c} = \mathbf{0}_{1\times K-M}$ which implies that (7) is true.

From (6), the L transmitted messages can be written in matrix form as

$$\mathbf{x}_{L\times 1} = \tilde{\mathbf{N}}_{\mathbf{D},\mathcal{M}}\ \mathbf{D}[f_1(W_1), f_2(W_2), \ldots, f_K(W_K)]^\intercal. \quad (8)$$

Since rank$(\tilde{\mathbf{N}}_{\mathbf{D},\mathcal{M}}) = L$, the matrix $\tilde{\mathbf{N}}_{\mathbf{D},\mathcal{M}}$ is invertible, and thus the user can decode the requested functions from $\mathbf{x}$ as

$$\mathbf{D}[f_1(W_1), f_2(W_2), \ldots, f_K(W_K)]^\intercal = (\tilde{\mathbf{N}}_{\mathbf{D},\mathcal{M}})^{-1}\mathbf{x}_{L\times 1}.$$

To show the achievability of $R_1(K, L, M) = L$ for any given $\mathbf{D} \in \mathbb{F}^{L\times K}$ with $K = L + M - 1$, we give an explicit construction of $\mathcal{M} = \{\mathcal{M}_1, \mathcal{M}_2, \ldots, \mathcal{M}_L\}$ satisfying the condition rank$(\mathbf{N}_{\mathbf{D},\mathcal{M}}) = L$. Without loss of generality, we assume that rank$(\mathbf{D}) = L$. From the full-row-rank matrix $\mathbf{D}$, we find an invertible submatrix $\mathbf{D}_\mathcal{L}$ of size $L \times L$, where $\mathcal{L} = \{i_1, i_2, \ldots, i_L\}$. Then, we form the task assignment set

$$\mathcal{M}^\star = \{\mathcal{M}^\star_1, \mathcal{M}^\star_2, \ldots, \mathcal{M}^\star_L\} \quad (9)$$

where, for every $\ell \in [L]$

$$\mathcal{M}^\star_\ell = \{i_\ell\} \bigcup ([K] \setminus \mathcal{L}). \quad (10)$$

Note that, $|\mathcal{M}^\star_\ell| = 1 + K - L = M$ for every $\ell \in [L]$. To complete the proof of achievability of the rate $R_1(K, L, M) = L$, it remains to verify that rank$(\mathbf{N}_{\mathbf{D},\mathcal{M}^\star}) = L$. Suppose rank$(\mathbf{N}_{\mathbf{D},\mathcal{M}^\star}) < L$, then there exists a row $\mathbf{N}_{\mathbf{D},\mathcal{M}^\star}(\ell', :)$ of $\mathbf{N}_{\mathbf{D},\mathcal{M}^\star}$ which can be represented as

$$\mathbf{N}_{\mathbf{D},\mathcal{M}^\star}(\ell', :) = \sum_{\ell=1,\ell\neq\ell'}^{L} \alpha_\ell \mathbf{N}_{\mathbf{D},\mathcal{M}^\star}(\ell, :) \quad (11)$$

where $\alpha_\ell \neq 0$ for some ℓ. On one hand, $\mathbf{N}_{\mathbf{D},\mathcal{M}^\star}(\ell', :)$ is a vector in the left nullspace of $\mathbf{D}_{\mathcal{M}^c_{\ell'}} = \mathbf{D}_{\mathcal{L}\setminus\{i_{\ell'}\}}$, and thus

$$\mathbf{N}_{\mathbf{D},\mathcal{M}^\star}(\ell', :)\mathbf{D}_\mathcal{L} = \beta_{\ell'}\mathbf{e}^\intercal_{\ell'} \quad (12)$$

where $\beta_{\ell'} \in \mathbb{F}\setminus\{0\}$ and $\mathbf{e}_{\ell'} \in \mathbb{F}^{L\times 1}$ is an L-length vector with a 1 in the ℓ'-th position and 0-s elsewhere. On the other hand, from (11), we have

$$\mathbf{N}_{\mathbf{D},\mathcal{M}^\star}(\ell', :)\mathbf{D}_\mathcal{L} = \left(\sum_{\ell=1,\ell\neq\ell'}^{L} \alpha_\ell \mathbf{N}_{\mathbf{D},\mathcal{M}^\star}(\ell, :)\right)\mathbf{D}_\mathcal{L}$$
$$= \sum_{\ell=1,\ell\neq\ell'}^{L} \alpha_\ell \Big(\mathbf{N}_{\mathbf{D},\mathcal{M}^\star}(\ell, :)\mathbf{D}_\mathcal{L}\Big) = \sum_{\ell=1,\ell\neq\ell'}^{L} \alpha_\ell \left(\beta_\ell \mathbf{e}^\intercal_\ell\right) \quad (13)$$

where $\beta_\ell \in \mathbb{F}\setminus\{0\}$, for every $\ell \in [L]\setminus\{\ell'\}$. Clearly, (12) and (13) contradict. Therefore, $\mathbf{N}_{\mathbf{D},\mathcal{M}^\star}$ cannot have rank less than L. In other words, rank$(\mathbf{N}_{\mathbf{D},\mathcal{M}^\star}) = L$, and therefore, for any given K, L, and M with $K = L + M - 1$, and for any $\mathbf{D} \in \mathbb{F}^{L\times K}$, the rate $R_1(K, L, M) = L$ is achievable. The optimality follows from the fact that to satisfy L linearly independent demands $R^*(K, L, M) \geq L$. This completes the proof of Lemma 1. □

We provide an example of Scheme 1 (Case 1) and demonstrate its achievable rate before proceeding to Case 2.

Example 1 (Scheme 1, Case 1)**.** *Consider the* $(K = 4, L = 3, M = 2)$ *distributed linearly separable function computation problem, where the demand matrix is*

$$\mathbf{D} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 2 & 3 & 2 \\ 1 & 4 & 1 & 2 \end{bmatrix}. \quad (14)$$

We now design the task assignment and the server transmissions based on Lemma 1. The number of servers required to employ the task assignment in Lemma 1 (cf. (5)*) is* $N = L = 3$*. Note that the first three columns of* $\mathbf{D}$ *are linearly independent. Thus, following the nullspace-based approach outlined in* (5)*, we get* $\mathcal{M}_1 = \{1, 4\}, \mathcal{M}_2 = \{2, 4\}$*, and* $\mathcal{M}_3 = \{3, 4\}$*. 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* $\mathbf{D}_{\mathcal{M}^c_1} = \mathbf{D}_{\{2,3\}}$*. For instance,* $[10, -3, -1]$ *is a vector in the left nullspace of* $\mathbf{D}_{\{2,3\}}$*. Then, the transmission made by the first server is*

$$\mathbf{x}_1 = [10, -3, -1][F_1(\mathcal{W}), F_2(\mathcal{W}), F_3(\mathcal{W})]^\intercal$$
$$= 6f_1(W_1) + 2f_4(W_4). \quad (15)$$

Similarly, the vectors $[-1, 0, 1]$ *and* $[-2, 3, -1]$ *are in the left nullspaces of* $\mathbf{D}_{\mathcal{M}^c_2} = \mathbf{D}_{\{1,3\}}$ *and* $\mathbf{D}_{\mathcal{M}^c_3} = \mathbf{D}_{\{1,2\}}$*, respectively. In the interest of space,* $\mathbf{x}_2$ *and* $\mathbf{x}_3$ *are omitted. Since the matrix constructed using* (4)

$$\mathbf{N}_{\mathbf{D},\mathcal{M}} = \begin{bmatrix} 10 & -3 & -1 \\ -1 & 0 & 1 \\ -2 & 3 & -1 \end{bmatrix}$$

is invertible, the user can decode $F_\ell(\mathcal{W})$ *for* $\ell = 1, 2, 3$*, from* $\mathbf{x}_1, \mathbf{x}_2$*, and* $\mathbf{x}_3$*. Thus, the rate achieved is* $R_1(K = 4, L = 3, M = 2) = 3$*. Since rank$_q(\mathbf{D}) = 3$, even a centralized server with* $M = 4$*, having access to all the datasets, would require* 3 *transmissions to meet the user's requests. Therefore, the optimal rate of the* $(K = 4, L = 3, M = 2)$ *distributed linearly separable function computation is* $R^*(4, 3, 2) = R_1(4, 3, 2) = 3$.

We now proceed to Case 2. The nullspace-based condition in Lemma 1 guarantees L linearly independent vectors, each with support size at most M, in the rowspace of $\mathbf{D}$. When $K > L + M - 1$, however, for any $\mathcal{M}_n \subseteq [K]$ with $|\mathcal{M}_n| \leq M$, the submatrix $\mathbf{D}_{\mathcal{M}^c_n}$ has at least $K - M$ columns, where $K - M > L - 1$. Hence, $\mathbf{D}_{\mathcal{M}^c_n}$ need not admit a non-trivial left nullspace. To combat this, we develop two techniques. First, we partition the demand matrix column-wise into sub-demand matrices and apply the method of Lemma 1 within each sub-demand matrix. Second, in Theorem 2, we introduce an approach that builds nullspaces by carefully augmenting the rows of the demand matrix.

2) ***Case 2*** *($K > L + M - 1$):*
Let $K' = L + M - 1$. We partition the demand matrix $\mathbf{D} \in \mathbb{F}^{L\times K}$ by column-wise into $\lceil K/K' \rceil$ sub-demand matrices as $\mathbf{D} = [\mathbf{D}_1, \mathbf{D}_2, \ldots, \mathbf{D}_{\lceil K/K' \rceil}]$, where $\mathbf{D}_\nu \in \mathbb{F}^{L\times K'}$ for all $\nu \in [\lceil K/K' \rceil - 1]$, and $\mathbf{D}_{\lceil K/K' \rceil} \in \mathbb{F}^{L\times (K - K'(\lceil K/K' \rceil - 1))}$.

Since the sub-demand matrix $\mathbf{D}_\nu$, $\nu \in [\lceil K/K' \rceil - 1]$ has K' (the number of columns in $\mathbf{D}_{\lceil K/K' \rceil}$ is less than or equal to K'), the coding scheme described in Lemma 1 applies, and is employed for $\mathbf{D}_\nu$. This coding strategy is applied separately for every $\nu \in [\lceil K/K' \rceil]$ using distinct sets of L servers. Thus, the user can retrieve the demanded functions at a rate

$$R_1(K, L, M) = L\left\lceil \frac{K}{L+M-1} \right\rceil \tag{16}$$

using $N = L\left\lceil \frac{K}{L+M-1} \right\rceil$ servers. □

We show the order optimality of Scheme 1 in the extended version [23]. In Scheme 1, after establishing the task assignment, our delivery scheme follows [19] in designing the nullspaces that govern transmission. Although the nullspaces differ from [19], the approach is similar, and in our setting it guarantees decodability for all demand matrices. Scheme 2 adds a careful augmentation of the demand matrix to optimize the sizes of the associated nullspaces. We now proceed with Scheme 2.

B. Scheme 2: Alternative Scheme Better Suited for Larger M

In this section, we propose another technique for building a target nullspace and satisfying the nullspace condition in Lemma 1 by augmenting the given demand matrix. The proposed technique is applicable for large values of M, specifically when $M \geq K/2$. Since we already have an optimal scheme for the case $L > K - M$ (Case 1 in Scheme 1), we focus on the scenario where $L \leq K - M$, and we have the following theorem.

Theorem 2. *For the (K, L, M) distributed linearly separable function computation problem with $L < \lfloor K/(K-M) \rfloor$, the rate*

$$R_2(K, L, M) = L + 1 \tag{17}$$

is achievable with $N = L + 1$ servers. Furthermore, for any $L \leq K - M$, the rate

$$R_2(K, L, M) = L + \left\lceil \frac{L}{\left\lfloor \frac{M}{K-M} \right\rfloor} \right\rceil \tag{18}$$

is achievable with $N = \lfloor K/(K-M) \rfloor$, if $M \geq K/2$.

Proof. Let $\mathbf{D} \in \mathbb{F}^{L \times K}$ be the demand matrix. First, we consider the case $L < \tau$, where $\tau = \lfloor K/(K-M) \rfloor = 1 + \lfloor M/(K-M) \rfloor$ is a positive integer. We set the number of servers $N = L + 1 \leq \tau$. To design the task assignment $\mathcal{M} = \{\mathcal{M}_1, \mathcal{M}_2, \ldots, \mathcal{M}_N\}$, we first partition the set $[K]$ as $\mathcal{P} = \{\mathcal{P}_1, \mathcal{P}_2, \ldots, \mathcal{P}_\tau\}$, where for every $t \in [\tau - 1]$, we define

$$\mathcal{P}_t = \{(t-1)(K-M)+1, (t-1)(K-M)+2, \ldots, t(K-M)\} \tag{19}$$

and

$$\mathcal{P}_\tau = \{(\tau-1)(K-M)+1, (\tau-1)(K-M)+2, \ldots, K\} \tag{20}$$

Note that, $|\mathcal{P}_t| = K - M$, for every $t \in [\tau - 1]$, and $|\mathcal{P}_\tau| = K - (\tau - 1)(K - M) \geq K - M$. which means that the task assignment set $\mathcal{M}$ takes the form

$$\mathcal{M} = \{\mathcal{P}_1^c, \mathcal{P}_2^c, \ldots, \mathcal{P}_N^c\}. \tag{21}$$

Note that $|\mathcal{M}_n| \leq M$, for every $n \in [N]$. Now, our objective is to design the server transmissions using the nullspace approach presented in Lemma 1. However, the left nullspace of $\mathbf{D}_{\mathcal{M}_n^c}$, for any $n \in [N]$, does not need to have a non-trivial vector, since $L \leq K - M$. Considering that, we augment the demand matrix by an additional row such that the augmented demand matrix, denoted by $\tilde{\mathbf{D}}$, satisfies the required condition in (5). Then, we have $\tilde{\mathbf{D}} = \begin{bmatrix} \mathbf{D} \\ \tilde{\mathbf{d}} \end{bmatrix} \in \mathbb{F}^{(L+1) \times K}$ where the construction of $\tilde{\mathbf{d}} \in \mathbb{F}^{1 \times K}$ is explained in the sequel. First, we define a matrix

$$\mathbf{T} = \begin{bmatrix} \mathbf{I}_{(N-1)\times(N-1)} & \mathbf{1}_{(N-1)\times 1} \\ \mathbf{0}_{1\times(N-1)} & 1 \end{bmatrix} \in \mathbb{F}^{N \times N} \tag{22}$$

which we refer to as the 'target nullspace matrix'. Note that $\text{rank}(\mathbf{T}) = N$ irrespective of the underlying field $\mathbb{F}$. We now construct the vector $\tilde{\mathbf{d}}$ as a concatenation of several vectors, where $\tilde{\mathbf{d}} = [\tilde{\mathbf{d}}_1, \tilde{\mathbf{d}}_2, \ldots, \tilde{\mathbf{d}}_N]$ if $\tau = \frac{K}{K-M}$ (when $(K - M)$ divides K) and $L = \tau - 1$. In that case, $\tilde{\mathbf{d}}_n \in \mathbb{F}^{1\times(K-M)}$, for every $n \in [N]$. If either $\tau < \frac{K}{K-M}$ or $L < \tau - 1$, we have $\tilde{\mathbf{d}} = [\tilde{\mathbf{d}}_1, \tilde{\mathbf{d}}_2, \ldots, \tilde{\mathbf{d}}_N, \tilde{\mathbf{d}}_{N+1}]$, where $\tilde{\mathbf{d}}_n \in \mathbb{F}^{1\times(K-M)}$, for every $n \in [N]$, and $\tilde{\mathbf{d}}_{N+1} \in \mathbb{F}^{1\times K-N(K-M)}$. For every $n \in [N-1]$, we now define

$$\tilde{\mathbf{d}}_n = -\mathbf{D}_{\mathcal{P}_n}(n, :) \text{ and } \tilde{\mathbf{d}}_N = \mathbf{0}_{1\times(K-M)}. \tag{23}$$

Furthermore, $\tilde{\mathbf{d}}_{N+1}$ can be set to any row vector of length $K - N(K-M)$. Now, using Lemma 1 on the augmented demand matrix $\tilde{\mathbf{D}}$ and the task assignment $\mathcal{M}$ in (21), we show the achievability of the rate $R_2 = N = L + 1$. That is, we show that $\text{rank}(\mathbf{N}_{\tilde{\mathbf{D}},\mathcal{M}}) = L + 1$, where

$$\mathbf{N}_{\tilde{\mathbf{D}},\mathcal{M}} = [\mathcal{N}(\tilde{\mathbf{D}}^{\intercal}_{\mathcal{M}_1^c}), \mathcal{N}(\tilde{\mathbf{D}}^{\intercal}_{\mathcal{M}_2^c}), \ldots, \mathcal{N}(\tilde{\mathbf{D}}^{\intercal}_{\mathcal{M}_N^c})]^{\intercal}.$$

Recall that, for every $n \in [N]$, we have $\mathcal{M}_n^c = \mathcal{P}_n$, and thus $\tilde{\mathbf{D}}_{\mathcal{M}_n^c} = \tilde{\mathbf{D}}_{\mathcal{P}_n}$. However, in the submatrix $\tilde{\mathbf{D}}_{\mathcal{P}_n}$, we have

$$\tilde{\mathbf{D}}_{\mathcal{P}_n}(n, :) + \tilde{\mathbf{D}}_{\mathcal{P}_n}(N, :) = \tilde{\mathbf{D}}_{\mathcal{P}_n}(n, :) + \tilde{\mathbf{d}}_n = \mathbf{0}_{1\times(K-M)} \tag{24}$$

for every $n \in [N-1]$, where (24) follows from the construction of $\tilde{\mathbf{d}}_n$ in (23). Therefore, for every $n \in [N]$, the vector $\mathbf{e_n} + \mathbf{e_N} \in \text{span}(\mathcal{N}(\tilde{\mathbf{D}}^{\intercal}_{\mathcal{M}_n^c}))$ (Note, $\mathbf{e_n}$ and $\mathbf{e_N} \in \mathbb{F}^{N\times 1}$). Notice that, we also have $\mathbf{e_n} + \mathbf{e_N} = \mathbf{T}(n, :)$, $\forall n \in [N-1]$. Therefore, the vector $\mathbf{T}(n, :)$, $n \in [N-1]$, is a row in $\mathbf{N}_{\tilde{\mathbf{D}},\mathcal{M}}$. Furthermore, in the submatrix $\tilde{\mathbf{D}}_{\mathcal{M}_N^c} = \tilde{\mathbf{D}}_{\mathcal{P}_N}$, we have $\tilde{\mathbf{D}}_{\mathcal{P}_N}(N, :) = \tilde{\mathbf{d}}_N = \mathbf{0}$. Thus, the vector $\mathbf{e_N} \in \text{span}(\mathcal{N}(\tilde{\mathbf{D}}^{\intercal}_{\mathcal{M}_N^c}))$, and note that $\mathbf{e_N} = \mathbf{T}(N, :)$. Consequently, the vector $\mathbf{T}(N, :)$ is a row in $\mathbf{N}_{\tilde{\mathbf{D}},\mathcal{M}}$. Therefore, we have $\text{rank}(\mathbf{N}_{\tilde{\mathbf{D}},\mathcal{M}}) \geq \text{rank}(\mathbf{T}) = N = L+1$. However, the number of columns in $\mathbf{N}_{\tilde{\mathbf{D}},\mathcal{M}}$ is N. Thus, the rank cannot exceed N. Therefore,

$$\text{rank}(\mathbf{N}_{\tilde{\mathbf{D}},\mathcal{M}}) = L + 1. \tag{25}$$

The achievability of $R_2(K, L, M) = L+1$ follows from (25).

In order to show the achievability of the expression in (18) for $L \geq \lfloor K/(K-M) \rfloor$, with $N = \lfloor K/(K-M) \rfloor$, we partition the demand matrix row-wise, and apply the technique presented in the first part of this proof on each of the sub-demand matrices. We have

$$\mathbf{D} = [\mathbf{D}_1; \mathbf{D}_2; \ldots; \mathbf{D}_{\lceil L/L' \rceil}]^{\intercal}$$

where $L' = \lfloor K/(K-M) \rfloor - 1 = \lfloor M/(K-M) \rfloor$. Since $L' < \lfloor K/(K-M) \rfloor$, the task assignment and transmissions based on a target nullspace matrix, presented in the first part of this proof, are applicable in each sub-demand matrix $\mathbf{D}_\lambda$, $\lambda \in [\lceil L/L' \rceil]$. Note that the task assignment set $\mathcal{M}$ is independent of $\mathbf{D}_\lambda$, and we have $\mathcal{M} = \{\mathcal{P}_1^c, \mathcal{P}_2^c, \ldots, \mathcal{P}_N^c\}$, $N = L'+1 = \lfloor K/(K-M) \rfloor$, where the sets $\mathcal{P}_n, n \in [N]$, are defined in (19) and (20). For each $\mathbf{D}_\lambda$, the transmission of the servers can be designed using the target nullspace matrix. The functions corresponding to each $\mathbf{D}_\lambda$, $\lambda \in [\lceil L/L' \rceil]$ can thus be decoded by the user. Corresponding to every $\mathbf{D}_\lambda$, $\lambda \in [\lceil L/L' \rceil - 1]$, each server makes a transmission, and corresponding to $\mathbf{D}_{\lceil L/L' \rceil}$, the first $L-(\lceil L/L' \rceil -1)L'+1$ servers make one transmission each and the remaining $L'\lceil L/L' \rceil - L$ servers do not make any transmission. Therefore, we get

$$R_2(K,L,M) = N\left(\left\lceil \frac{L}{L'} \right\rceil - 1\right) + L - (\lceil L/L' \rceil - 1)L' + 1$$
$$= L + \left\lceil \frac{L}{L'} \right\rceil = L + \left\lceil \frac{L}{\left\lfloor \frac{M}{K-M} \right\rfloor} \right\rceil .$$

This completes the proof of Theorem 2. □

Example 2 (Scheme 2). *Consider the* $(K=6, L=2, M=4)$ *distributed linearly separable function computation problem with a demand matrix*

$$\mathbf{D} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 2 & 3 & 4 & 5 \end{bmatrix}.$$

Note that $L < K/(K-M) = 3$*. From the scheme described in the proof of Theorem 2, we have* $N = L+1 = 3$*. The task assignment on the three servers is* $\mathcal{M}_1 = \{3,4,5,6\}, \mathcal{M}_2 = \{1,2,5,6\}$*, and* $\mathcal{M}_3 = \{1,2,3,4\}$*. Then the augmented demand matrix* $\tilde{\mathbf{D}}$ *is constructed in such a way that the vector* $\mathbf{T}(1,:) = [1,0,1]$ *falls in the left nullspace of the submatrix* $\tilde{\mathbf{D}}_{\{1,2\}}$*. Similarly, the vectors* $[0,1,1]$ *and* $[0,0,1]$ *should be in the left nullspaces of* $\tilde{\mathbf{D}}_{\{3,4\}}$ *and* $\tilde{\mathbf{D}}_{\{5,6\}}$*, respectively. Therefore, we get* $\tilde{\mathbf{D}} = \begin{bmatrix} \mathbf{D} \\ \tilde{\mathbf{d}} \end{bmatrix}$*, where* $\tilde{\mathbf{d}} = [-1,-1,-2,-3,0,0]$*. Using* $\tilde{\mathbf{D}}$ *and* $\mathbf{T}$*, each of the three servers makes a transmission, and the user can decode the demanded functions from* $\mathbf{x}_1, \mathbf{x}_2$*, and* $\mathbf{x}_3$*. Due to space constraints, the transmissions are not listed here. Thus, the rate achieved is* $R_2(6,2,4) = 3$*.*

Remark 1. *In Example 2, instead of the specific row added, any random third row would suffice to achieve the rate* $R_1(6,2,4) = 3$ *using Scheme 1. This is because for* $L = 3$ *and* $M = 4$*, we have* $K = L+M-1 = 6$*, and Lemma 1 guarantees achievability of* $R_1(6,2,4) = 3$*. However, this equivalence does not hold in general. For example, when* $K = 9$*,* $M = 6$*, and* $L = 2$*, Scheme 2 achieves a communication cost of* $R_2 = 3$*, whereas Scheme 1 requires two additional random rows to satisfy* $K = L+M-1$*, leading to a higher cost of* $R_1 = 4$*.*

IV. Conclusion

In this work, we proposed distributed computing schemes for linearly separable functions. Our schemes are based on a nullspace-based design principle that jointly determines the task assignment and transmission policies while guaranteeing exact decodability of the requested functions. Furthermore, we showed that the optimal communication cost L is achievable when $K < L+M$. Identifying the fundamental limits of this distributed computing problem and extending the performance guarantees of the proposed schemes remain part of our ongoing work.

Acknowledgement

This research was partially supported by European Research Council ERC-StG Project SENSIBILITE under Grant 101077361, the (ERC)-PoC Project LIGHT under Grant 101101031, by the Huawei France-Funded Chair Toward Future Wireless Networks, and in part by the Program "PEPR Networks of the Future" of France 2030.

References

[1] J. Dean and S. Ghemawat, "MapReduce: Simplified data processing on large clusters," *Commun. ACM*, vol. 51, no. 1, p. 107–113, Jan. 2008.

[2] M. Zaharia, M. Chowdhury, M. J. Franklin, S. Shenker, and I. Stoica, "Spark: Cluster computing with working sets," in *Proc. USENIX Conf. Hot Topics in Cloud Comput.*, ser. HotCloud'10, 2010, p. 10.

[3] 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.

[4] 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.

[5] 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.

[6] 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.

[7] D. Malak, M. R. Deylam Salehi, B. Serbetci, and P. Elia, "Multi-server multi-function distributed computation," *Entropy*, vol. 26, no. 6, 2024.

[8] M. Xhemrishi, R. Bitar, and A. Wachter-Zeh, "Distributed matrix-vector multiplication with sparsity and privacy guarantees," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, 2022, pp. 1028–1033.

[9] Q. Yu and A. S. Avestimehr, "Coded computing for resilient, secure, and privacy-preserving distributed matrix multiplication," *IEEE Trans. Commun.*, vol. 69, no. 1, pp. 59–72, 2021.

[10] 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.

[11] 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.

[12] E. Parrinello, E. Lampiris, 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.

[13] F. Brunero and P. Elia, "Multi-access distributed computing," *IEEE Trans. Inf. Theory*, vol. 70, no. 5, pp. 3385–3398, 2024.

[14] 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.

[15] F. Brunero, K. Wan, G. Caire, and P. Elia, "Coded distributed computing for sparse functions with structured support," in *Proc. IEEE Inf. Theory Workshop (ITW)*, 2023, pp. 474–479.

[16] M. Ye and E. Abbe, "Communication-computation efficient gradient coding," in *Proc. Int. Conf. Mach. Learn. (ICML)*, 2018, pp. 5610–5619.

[17] 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)*, 2024, pp. 816–820.

[18] A. Ramamoorthy, L. Tang, and P. O. Vontobel, "Universally decodable matrices for distributed matrix-vector multiplication," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, 2019, pp. 1777–1781.

[19] K. Wan, H. Sun, M. Ji, and G. Caire, "Distributed linearly separable computation," *IEEE Trans. Inf. Theory*, vol. 68, no. 2, pp. 1259–1278, 2022.

[20] ——, "On the tradeoff between computation and communication costs for distributed linearly separable computation," *IEEE Trans. Commun.*, vol. 69, no. 11, pp. 7390–7405, 2021.

[21] A. Khalesi and P. Elia, "Multi-user linearly-separable distributed computing," *IEEE Trans. Inf. Theory*, vol. 69, no. 10, pp. 6314–6339, 2023.

[22] ——, "Tessellated distributed computing," *IEEE Trans. Inf. Theory*, vol. 71, no. 6, pp. 4754–4784, 2025.

[23] K. K. K. Namboodiri, E. Peter, D. Malak, and P. Elia, "Fundamental limits of distributed computing for linearly separable functions," 2025. [Online]. Available: https://arxiv.org/abs/2509.23447