Fundamental Limits of Distributed Linearly Separable Computation under Cyclic Assignment

Wenbo Huang*, Kai Wan*, Hua Sun†, Mingyue Ji‡, Robert Caiming Qiu*, Giuseppe Caire§
*Huazhong University of Science and Technology, 430074 Wuhan, China, {eric_huang,kai_wan,caiming}@hust.edu.cn
†University of Utah, Salt Lake City, UT 84112, USA, mingyue.ji@utah.edu
‡University of North Texas, Denton, TX 76203, USA, hua.sun@unt.edu
§Technische Universität Berlin, 10587 Berlin, Germany, caire@tu-berlin.de

2023 IEEE International Symposium on Information Theory (ISIT) | 978-1-6654-7554-9/23/$31.00 ©2023 IEEE | DOI: 10.1109/ISIT54713.2023.10206661

***Abstract*—Distributed Linearly Separable Computation problem under the cyclic assignment is studied in this paper. It is a problem widely existing in cooperated distributed gradient coding, real-time rendering, linear transformers, etc. In a distributed computing system, a master asks N distributed workers to compute a linearly separable function from K datasets. The task function can be expressed as K_c linear combinations of K messages, where each message is the output of one individual function of one dataset. Straggler effect is also considered, such that from the answers of each N_r worker, the master should recover the task. The computation cost is defined as the number of datasets assigned to each worker, while the communication cost is defined as the number of (coded) messages which should be received. The objective is to characterize the optimal tradeoff between the computation and communication costs. Various distributed computing scheme were proposed in the literature with a well-known cyclic data assignment, but the (order) optimality of this problem remains open, even under the cyclic assignment. This paper proposes a new computing scheme with the cyclic assignment based on interference alignment, which is near optimal under the cyclic assignment.**

***Index Terms*—Coded distributed computing, linearly separable function, cyclic assignment, interference alignment**

I. Introduction

With the development of deep learning and communication technologies, computation on large-scale data is an emergent challenge to be solved [1]. To economize the resource of computers and speed up the computing process, clients always adopt distributed computing technology in the cloud platform to carry out large and complex tasks [2]. The cloud computing platforms such as Amazon Web Services (AWS) [3], Microsoft Azure [4] and Google Cloud Platform [5] are widely used in real systems. Some distributed computing framework like Apache Spark [6], MapReduce [7] attract a lot of cutting-edge research [8]–[10]. However, the performance of a distributed system is strongly affected by the *straggling workers* (or simply, *stragglers*) and the limited communication bandwidth [11]. Coding techniques (such as error-correction codes and linear network codes) were originally introduced to efficiently solve the above two challenges in the distributed matrix multiplication problem [11] and in the MapReduce distributed computing problem [12].

This paper considers a specific distributed computing framework, distributed linearly separable computation, originally proposed in [13], [14]. A master wants to compute a function of K datasets with equal length expressed as K_c linear combinations of K messages, where each message is an individual function of one distinct dataset. This computation task structure covers many practical applications as special cases, such as distributed gradient descent [15], distributed linear transform [16], real-time rendering [17], etc. In the system considered in this paper, the master asks N workers to compute a linearly separable function on K datasets $(D_1, \ldots, D_\mathsf{K})$:[1]

$$f(D_1, \ldots, D_\mathsf{K}) = g(f_1(D_1), \ldots, f_\mathsf{K}(D_\mathsf{K})) = g(W_1, \ldots, W_\mathsf{K}),$$

where $W_k = f_k(D_k), k \in [\mathsf{K}]$. is the result of the computation when the dataset D_k applying to function $f_k(\cdot)$ and $W_k \in \mathbb{F}_\mathsf{q}^\mathsf{L}$. $g(W_1, \ldots, W_\mathsf{K})$ contains K_c linear combinations of the messages $W_1, \ldots, W_\mathsf{K}$, where each message contains L symbols uniformly i.i.d. in $\mathbb{F}_\mathsf{q}$. In a matrix form, $g(W_1, \ldots, W_\mathsf{K}) = \mathbf{F}[W_1; \ldots; W_\mathsf{K}]$, where $\mathbf{F}$ is a demand matrix with dimension $\mathsf{K}_\mathrm{c} \times \mathsf{K}$ whose elements are uniformly i.i.d. in $\mathbb{F}_\mathsf{q}$. The distributed computing framework contains three phases:

- *Assignment phase.* The master assigns the datasets to the workers in an uncoded way. Each worker receives $\mathsf{M} = \frac{\mathsf{K}}{\mathsf{N}}(\mathsf{N} - \mathsf{N}_\mathrm{r} + \mathsf{m})$ datasets, where $m \in \{1, \ldots, \mathsf{N}_\mathrm{r}\}$.
- *Computing phase.* Each worker n where $n \in \{1, \ldots, \mathsf{N}\}$ computes the messages from the assigned dataset, and then sends back coded messages to the master.
- *Decoding phase.* After receiving the answers of any N_r workers, the master should be able to recover the computing task; i.e., we should tolerate up to $\mathsf{N} - \mathsf{N}_\mathrm{r}$ stragglers.

The objective is to minimize the communication cost R, which is defined as the (normalized) number of symbols which should be received by the master in order to recover the computing task.

In the literature, various works have been proposed to consider different regimes of system parameters for the above

[1] As in [13], [14], we assume that K divides N in this paper. The proposed computing scheme can be extended to the case where K does not divide N, by adding virtual datasets as the step in [13], [14, Section V-A].

distributed linearly separable computation problem:[2]

- $\mathsf{m}=1$, $\mathsf{K_c}=1$. Computing schemes with the optimal communication cost were proposed in [15].
- $\mathsf{m}>1$, $\mathsf{K_c}=1$. Under the constraint of linear coding, computing schemes with the optimal communication cost were proposed in [18], [19].
- $\mathsf{m}=1$, $\mathsf{K_c}>1$. Under the constraint of the cyclic assignment, a computing scheme with the optimal communication cost was proposed in [13].
- $\mathsf{m}>1$, $\mathsf{K_c}>1$. Under the constraint of the cyclic assignment and $\mathsf{N} \geq \frac{\mathsf{m}+\mathsf{u}-1}{\mathsf{u}} + \mathsf{u}\,(\mathsf{N_r}-\mathsf{m}-\mathsf{u}+1)$ where $\mathsf{u} := \left\lceil \frac{\mathsf{K_c N}}{\mathsf{K}} \right\rceil$, a computing scheme was proposed in [14], which is optimal if $\mathsf{N}=\mathsf{K}$ and order optimal within a factor of 2 otherwise.

As a summary on the existing works on the distributed linearly separable computation problem, the (order) optimal communication cost remains open, even under the constraint of the cyclic assignment.

Contributions: For the distributed linearly separable computation problem with $\mathsf{m}>1$ and $\mathsf{K_c}>1$, we proposed a new computing scheme with the cyclic assignment and a computing phase based on interference alignment. Under the constraint of the cyclic assignment, the required communication cost of the proposed computing scheme is optimal if $\mathsf{N}=\mathsf{K}$ and order optimal within a factor of 2 otherwise.

Notations: Calligraphic symbols denote sets, bold symbols denote vectors and matrices, and sans-serif symbols denote system parameters. We use $|\cdot|$ to represent the cardinality of a set or the length of a vector; $[a:b] := \{a, a+1, \ldots, b\}$ and $[n] := [1:n]$; the sum of a set S and a scalar a represents the resulting set of adding each element of S by a; $\mathbf{M}^{\mathrm{T}}$ and $\mathbf{M}^{-1}$ represent the transpose and the inverse of matrix $\mathbf{M}$, respectively; $\mathbb{F}_{\mathsf{q}}$ represents a finite field with order q; in this paper, the basis of logarithm in the entropy terms is q; $\mathrm{Mod}(b,a)$ represents the modulo operation on b with integer divisor a and in this paper we let $\mathrm{Mod}(b,a) \in \{1,\ldots,a\}$ (i.e., we let $\mathrm{Mod}(b,a)=a$ if a divides b).

II. System Model

We consider the $(\mathsf{K},\mathsf{N},\mathsf{N_r},\mathsf{K_c},\mathsf{m})$ distributed linearly separable computation problem over the canonical master-worker distributed system, originally proposed in [13]. A master wants to compute a function of K datasets $D_1,\ldots,D_{\mathsf{K}}$, by the help of N workers.

With the assumption that the function is linearly separable from the datasets, the computation task can be written as $\mathsf{K_c} \leq \mathsf{K}$ linear combinations of K messages

$$f(D_1,D_2,\ldots,D_{\mathsf{K}}) = g(f_1(D_1),\ldots,f_{\mathsf{K}}(D_{\mathsf{K}}))$$
$$= g(W_1,\ldots,W_{\mathsf{K}}) = \mathbf{F}[W_1;\ldots;W_{\mathsf{K}}] = [F_1;\ldots;F_{\mathsf{K_c}}], \quad (1)$$

where the i^{th} message is $W_i = f_i(D_i)$, representing the outcome of the component function $f_i(\cdot)$ applied to dataset D_i. As in [13], we assume that each message W_i contains L uniformly i.i.d. symbols in $\mathbb{F}_{\mathsf{q}}$, where q is large enough.[3] $\mathbf{F}$ represents the demand matrix with dimension $\mathsf{K_c}\times\mathsf{K}$, where each of its elements is uniformly i.i.d. over $\mathbb{F}_{\mathsf{q}}$.

A distributed computing framework contains three phases.

Data assignment phase: We assign $\mathsf{M} := \frac{\mathsf{K}}{\mathsf{N}}(\mathsf{N}-\mathsf{N_r}+\mathsf{m})$ datasets to each worker. The set of indices of datasets assigned to worker n is denoted by $\mathcal{Z}_n$, where $\mathcal{Z}_n \subseteq [\mathsf{K}]$ and $|\mathcal{Z}_n| = \mathsf{M}$.

Computing phase: Each worker $n \in [\mathsf{N}]$ first computes the messages $W_k = f_k(D_k)$ for each $k \in \mathcal{Z}_n$. Then it computes X_n, which a function of the M messages $\{W_k : k \in \mathcal{Z}_n\}$, and sends X_n back to the master. The number of symbols in X_n is denoted by T_n.

Since the computation complexity on the separable functions is usually is usually much higher than computing the desired linear combinations of the messages, the computation cost of each worker is defined as M.

Decoding phase: The master only waits for the answers of the first $\mathsf{N_r}$ workers. Since the master and workers cannot foresee which $\mathsf{N_r}$ workers arrive first, the computing scheme should be designed to tolerate any $\mathsf{N}-\mathsf{N_r}$ stragglers. For each subset of workers $\mathcal{A} \subseteq [\mathsf{N}]$ where $|\mathcal{A}| = \mathsf{N_r}$, by defining $X_{\mathcal{A}} := \{X_n : n \in \mathcal{A}\}$, there should exist a decoding function such that $\hat{g}_{\mathcal{A}} = \phi_{\mathcal{A}}\big(X_{\mathcal{A}}, \mathbf{F}\big)$, where $\phi_{\mathcal{A}} : \mathbb{F}_{\mathsf{q}}^{|\mathcal{Z}_n|\mathsf{L}} \times [\Gamma_q]^{\mathsf{K_c}\times\mathsf{K}} \to [\Gamma_q]^{\mathsf{K_c}\times\mathsf{L}}$.

The worst-case error probability is defined as

$$\varepsilon := \max_{\mathcal{A}\subseteq[\mathsf{N}]:|\mathcal{A}|=\mathsf{N_r}} \Pr\{\hat{g}_{\mathcal{A}} \neq g(W_1,\ldots,W_{\mathsf{K}})\}. \quad (2)$$

A computing scheme is called achievable if the worst-case error probability vanishes when $\mathsf{q}\to\infty$.

We define

$$\mathsf{R} := \max_{\mathcal{A}\subseteq[\mathsf{N}]:|\mathcal{A}|=\mathsf{N_r}} \frac{\sum_{n\in\mathcal{A}} \mathsf{T}_n}{\mathsf{L}} \quad (3)$$

as the communication cost, which presents the worst-case (normalized) number of symbols received by the master from any $\mathsf{N_r}$ responding workers to recover the computation task.

Objective: The objective of the $(\mathsf{K},\mathsf{N},\mathsf{N_r},\mathsf{K_c},\mathsf{m})$ distributed linearly separable computation problem is to characterize the optimal (minimum) communication cost $\mathsf{R}^\star$ among all achievable computing schemes. Notice that, in order to tolerate $\mathsf{N}-\mathsf{N_r}$ stragglers, we should have $\mathsf{m}\in[\mathsf{N}-\mathsf{N_r}]$.

Cyclic assignment: Under the cyclic assignment, each dataset D_i, where $i \in [\mathsf{K}]$, is assigned to workers $\mathrm{Mod}(i,\mathsf{N}), \mathrm{Mod}(i-1,\mathsf{N}), \ldots, \mathrm{Mod}(i-\mathsf{M}+1,\mathsf{N})$. Thus for each worker $n\in[\mathsf{N}]$, we have

$$\mathcal{Z}_n = \bigcup_{p\in\left[0:\frac{\mathsf{K}}{\mathsf{N}}-1\right]} \big\{\mathrm{Mod}(n,\mathsf{N})+p\mathsf{N}, \mathrm{Mod}(n+1,\mathsf{N})+p\mathsf{N}, \ldots,$$
$$\mathrm{Mod}(n+\mathsf{N}-\mathsf{N_r}+\mathsf{m},\mathsf{N})+p\mathsf{N}\big\} \quad (4)$$

[2]The following mentioned existing schemes are all with a well-known cyclic assignment. The cyclic assignment was used in most existing works on the distributed linearly separable computation problem and also in practical systems. The main advantages of the cyclic assignment are: (i) it can be widely used unlimited by system parameters, (ii) its independence of the computing task such that the data assignment could be done offline, (iii) its simplicity.

[3]In this paper, we assume that K/N is an integer and L is large enough such that any sub-message division is possible.

with cardinality $\frac{\mathsf{K}}{\mathsf{N}}(\mathsf{N}-\mathsf{N}_r+\mathsf{m})$. The optimal communication cost under the cyclic assignment is defined as $\mathsf{R}^\star_{\text{cyc}}$.

A converse bound on $\mathsf{R}^\star_{\text{cyc}}$ was propsed in [14], which is reviewed as follows.

Theorem 1 ([14])**.** *For the* $(\mathsf{K},\mathsf{N},\mathsf{N}_r,\mathsf{K}_c,\mathsf{m})$ *distributed linearly separable computation problem,*

- *when* $\mathsf{K}_c \in \left[\frac{\mathsf{K}}{\mathsf{N}}(\mathsf{N}_r-\mathsf{m}+1)\right]$*, by defining* $\mathsf{u} := \left\lceil \frac{\mathsf{K}_c\mathsf{N}}{\mathsf{K}} \right\rceil$*, we have*
$$\mathsf{R}^\star_{cyc} \geq \frac{\mathsf{N}_r\mathsf{K}_c}{\mathsf{m}+\mathsf{u}-1}. \tag{5a}$$
- *when* $\mathsf{K}_c \in \left[\frac{\mathsf{K}}{\mathsf{N}}(\mathsf{N}_r-\mathsf{m}+1):\mathsf{K}\right]$*, we have*
$$\mathsf{R}^\star_{cyc} \geq \mathsf{R}^\star \geq \mathsf{K}_c. \tag{5b}$$

As explained in the Introduction, the order optimality under the constraint of the cyclic assignment still remains open.

III. Main Results

In this section, we provide a new achievable computing scheme, whose required communication cost is stated in the following theorem.

Theorem 2. *For the* $(\mathsf{K},\mathsf{N},\mathsf{N}_r,\mathsf{K}_c,\mathsf{m})$ *distributed linearly separable computation problem where* $\mathsf{N} \leq 60$*, the following communication cost is achievable, where*

- *when* $\mathsf{K}_c \in \left[\frac{\mathsf{K}}{\mathsf{N}}\right]$,
$$\mathsf{R}_1 = \frac{\mathsf{K}_c\mathsf{N}_r}{\mathsf{m}}; \tag{6}$$
- *when* $\mathsf{K}_c \in \left[\frac{\mathsf{K}}{\mathsf{N}}:\frac{\mathsf{K}}{\mathsf{N}}(\mathsf{N}_r-\mathsf{m}+1)\right]$,
$$\mathsf{R}_1 = \frac{\mathsf{N}_r\mathsf{K}\mathsf{u}}{\mathsf{N}(\mathsf{m}+\mathsf{u}-1)}; \tag{7}$$
- *when* $\mathsf{K}_c \in \left[\frac{\mathsf{K}}{\mathsf{N}}(\mathsf{N}_r-\mathsf{m}+1):\mathsf{K}\right]$,
$$\mathsf{R}_1 = \mathsf{K}_c. \tag{8}$$

Note that when $\mathsf{K}_c \in \left[\frac{\mathsf{K}}{\mathsf{N}}\right]$, to achieve $\mathsf{R}_1 = \frac{\mathsf{K}_c\mathsf{N}_r}{\mathsf{m}}$, we directly repeat the optimal computing schemes for $\mathsf{K}_c = 1$ and $\mathsf{m} > 1$ in [18], [19] K_c times. In addition, as explained in [13], if the proposed scheme works for the case $\mathsf{K}_c = \frac{\mathsf{K}}{\mathsf{N}}(\mathsf{N}_r-\mathsf{m}+1)$ with communication cost K_c, then it also works for the case $\mathsf{K}_c \in \left[\frac{\mathsf{K}}{\mathsf{N}}(\mathsf{N}_r-\mathsf{m}+1):\mathsf{K}\right]$ with communication cost K_c. The main contribution of our proposed computing scheme is for the case $\mathsf{K}_c \in \left[\frac{\mathsf{K}}{\mathsf{N}}+1:\frac{\mathsf{K}}{\mathsf{N}}(\mathsf{N}_r-\mathsf{m}+1)\right]$. We use the Schwartz-Zippel lemma [20]–[22] to prove the decodability of the proposed computing scheme in Theorem 2. For the non-zero polynomial condition for the Schwartz-Zippel lemma, for the following two cases (i) $\mathsf{K}_c \in \left[\frac{\mathsf{K}}{\mathsf{N}}(\mathsf{N}_r-\mathsf{m}+1):\mathsf{K}\right]$ and (ii) $\mathsf{N}=\mathsf{N}_r$, $\mathsf{m}=2$ and K_c+1 divides N, we provide formal proofs to show the non-zero polynomial condition, and thus prove our scheme is decodable. We also numerically verify all cases that $\mathsf{N} \leq 60$, and thus conjecture in the rest of the paper, **the proposed computing scheme is decodable for any system parameters** with the communication cost given in Theorem 2.

The proposed computing scheme fully covers the computing scheme in [14], which only works for the case $\mathsf{N} \geq \frac{\mathsf{m}+\mathsf{u}-1}{\mathsf{u}} + \mathsf{u}(\mathsf{N}_r-\mathsf{m}-\mathsf{u}+1)$.

By comparing the proposed scheme in Theorem 2 with the converse bound in Theorem 1, we obtain the following (order) optimality results.

Theorem 3. *For the* $(\mathsf{K},\mathsf{N},\mathsf{N}_r,\mathsf{K}_c,\mathsf{m})$ *distributed linearly separable computation problem,*

1) *when* $\mathsf{K}=\mathsf{N}$*, we have*
$$\mathsf{R}^\star_{cyc} = \mathsf{R}_1 = \begin{cases} \frac{\mathsf{N}_r\mathsf{K}_c}{\mathsf{m}+\mathsf{u}-1}, & \textit{if } \mathsf{K}_c \in [\mathsf{N}_r-\mathsf{m}+1]; \\ \mathsf{K}_c, & \textit{if } \mathsf{K}_c \in [\mathsf{N}_r-\mathsf{m}+1:\mathsf{K}]; \end{cases} \tag{9}$$
2) *when* $\mathsf{K}_c \in \left[\frac{\mathsf{K}}{\mathsf{N}}\right]$*, we have*
$$\mathsf{R}^\star_{cyc} = \mathsf{R}_1 = \frac{\mathsf{N}_r\mathsf{K}_c}{\mathsf{m}}; \tag{10}$$
3) *when* $\mathsf{K}_c \in \left[\frac{\mathsf{K}}{\mathsf{N}}+1:\frac{\mathsf{K}}{\mathsf{N}}(\mathsf{N}_r-\mathsf{m}+1)-1\right]$*, we have*
$$\mathsf{R}^\star_{cyc} \geq \frac{\mathsf{K}_c}{\frac{\mathsf{K}}{\mathsf{N}}\mathsf{u}}\mathsf{R}_1 \geq \frac{\mathsf{R}_1}{2}; \tag{11}$$
4) *when* $\mathsf{K}_c \in \left[\frac{\mathsf{K}}{\mathsf{N}}(\mathsf{N}_r-\mathsf{m}+1):\mathsf{K}\right]$*, we have*[4]
$$\mathsf{R}^\star = \mathsf{R}^\star_{cyc} = \mathsf{R}_1 = \mathsf{K}_c. \tag{12}$$

To summarize Theorem 3, under the constraint of the cyclic assignment, the proposed computing scheme is order optimal within a factor of 2, for all system parameters.

In the following, we will describe the proposed computing scheme in Theorem 2 for the case $\mathsf{K}_c \in \left[\frac{\mathsf{K}}{\mathsf{N}}+1:\frac{\mathsf{K}}{\mathsf{N}}(\mathsf{N}_r-\mathsf{m}+1)\right]$, to prove (7). By a similar proof as in [14, Appendix A], if the proposed scheme works for the $(\mathsf{N},\mathsf{N},\mathsf{N}_r,\mathsf{u},\mathsf{m})$ problem with high probability and achieves (7), then it can also be extended to the $\left(\mathsf{K},\mathsf{N},\mathsf{N}_r,\frac{\mathsf{K}}{\mathsf{N}}\mathsf{u},\mathsf{m}\right)$ problem with high probability and achieves (7). Hence, in the proof of Theorem 2, for the ease of description, we only consider the case where $\mathsf{K}=\mathsf{N}$. The proposed scheme differs in three regimes, $\mathsf{K}_c = \mathsf{N}_r-\mathsf{m}+1$, $\mathsf{K}_c=\mathsf{N}_r-\mathsf{m}$, and $\mathsf{K}_c \in [2:\mathsf{N}_r-\mathsf{m}-1]$. Due to the limitation of pages, we use one example to illustrate the main ideas of the proposed scheme for $\mathsf{K}_c \in [2:\mathsf{N}_r-\mathsf{m}-1]$, which is the most non-trivial scheme and is based on interference alignment.

Example 1 ($(\mathsf{K},\mathsf{N},\mathsf{N}_r,\mathsf{u},\mathsf{m}) = (6,6,6,2,2)$)**.** *We consider the example where* $\mathsf{N}=\mathsf{K}=6$*,* $\mathsf{N}_r=6$*,* $\mathsf{K}_c=\mathsf{u}=2$*, and* $\mathsf{M}=\mathsf{m}=2$*. The converse bound under the cyclic assignment in Theorem 1 for this example is* $\mathsf{R}^\star_{cyc} \geq \frac{\mathsf{N}_r\mathsf{K}_c}{\mathsf{m}+\mathsf{u}-1} = 4$*. Without loss general, we assume the demand matrix* $\mathbf{F}$ *is*
$$\mathbf{F} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 2 & 3 & 4 & 5 & 6 \end{bmatrix}. \tag{13}$$
For the sake of simplicity, in this example, we assume that the field is a large enough prime field; in general the proposed

[4]For the considered problem, it is natural to see that the optimal communication cost $\mathsf{R}^\star$ is lower bounded by K_c, since the K_c demanded linear combinations of messages are linearly independent with high probability.

scheme does not need this assumption (recall we only need the field size is large enough). Note that the computing scheme in [14] cannot work for this example.

Data assignment phase: We consider the cyclic assignment, which is illustrated in Table I.

TABLE I: Data assignment

worker 1	worker 2	worker 3	worker 4	worker 5	worker 6
D_1 D_2	D_2 D_3	D_3 D_4	D_4 D_5	D_5 D_6	D_6 D_1

Computing phase: To achieve the converse bound $\mathsf{R}^\star_{cyc} \geq \frac{\mathsf{N_r K_c}}{\mathsf{m+u-1}} = 4$, *we divide each message* W_k *where* $k \in [\mathsf{K}]$ *into* $\mathsf{d} := \mathsf{m}+\mathsf{u}-1 = 3$ *non-overlapping and equal-length sub-messages,* $W_k = \{W_{k,j} : j \in [\mathsf{d}]\}$, *where each sub-message contains* $\frac{\mathsf{L}}{\mathsf{d}} = \frac{\mathsf{L}}{3}$ *symbols in* $\mathbb{F}_\mathsf{q}$. *We will let each worker transmit* $\frac{\mathsf{K_c L}}{\mathsf{d}} = \frac{2\mathsf{L}}{3}$ *symbols in the computing scheme, which coincides with the converse bound. After the message division, the computation task becomes* $\mathsf{K_c} \times \mathsf{d} = 6$ *linear combinations of sub-messages. Since each worker transmits* 2 *linear combinations of sub-messages, the master totally receives* 12 *linear combinations of sub-messages, whose spanned linear space contains the computation task. In other words, the total received linear combinations contains* 6 *virtually demanded linear combinations of sub-messages. The effective demand could be expressed as* $\mathbf{F}' \begin{bmatrix} W_{1,1} \\ \vdots \\ W_{6,1} \\ W_{1,2} \\ \vdots \\ W_{6,3} \end{bmatrix} = \mathbf{F}'\mathbf{W}$, *where* $\mathbf{W} = \begin{bmatrix} W_{1,1} \\ \vdots \\ W_{6,1} \\ W_{1,2} \\ \vdots \\ W_{6,3} \end{bmatrix}$, *and the dimension of* $\mathbf{F}'$ *is* $\mathsf{N_r K_c} \times \mathsf{K}(\mathsf{m}+\mathsf{u}-1) = 12 \times 18$, *with the form*

$$\mathbf{F}' = \left[\begin{array}{c|c|c} (\mathbf{F})_{2\times 6} & 0_{2\times 6} & 0_{2\times 6} \\ \hline 0_{2\times 6} & (\mathbf{F})_{2\times 6} & 0_{2\times 6} \\ \hline 0_{2\times 6} & 0_{2\times 6} & (\mathbf{F})_{2\times 6} \\ \hline (\mathbf{V}_1)_{6\times 6} & (\mathbf{V}_2)_{6\times 6} & (\mathbf{V}_3)_{6\times 6} \end{array}\right], \tag{14}$$

where $(\mathbf{V_i})_{6\times 6}, i \in [3]$ *are denoted as the virtual demands.*

Our coding strategy is to let each worker $n \in [6]$ *send two linear combinations*

$$\mathbf{s}^{n,1}\mathbf{F}'\mathbf{W} \text{ and } \mathbf{s}^{n,2}\mathbf{F}'\mathbf{W}, \tag{15}$$

where $\mathbf{s}^{n,1}$ *and* $\mathbf{s}^{n,2}$ *are vectors with length* 12. *By the cyclic assignment, worker* n *does not have* $\{D_k : k \in \overline{\mathcal{Z}_n}\}$ *where* $\overline{\mathcal{Z}_n} = [6] \setminus \{n, Mod(n+1, 6)\}$. *Consequently it cannot compute the sub-messages* $\{W_{k,j} : k \in \overline{\mathcal{Z}_n}, j \in [3]\}$, *and thus in the transmitted linear combinations* (15) *the coefficients of the sub-message which it cannot compute are* 0*; i.e., the* i^{th} *columns of* $\mathbf{s}^{n,1}\mathbf{F}'$ *and of* $\mathbf{s}^{n,2}\mathbf{F}'$ *are* 0, *where* $i \in \mathcal{Z}_n \cup (\mathcal{Z}_n + 6) \cup (\mathcal{Z}_n + 12)$. *Hence, the column-wise sub-matrix of* $\mathbf{F}'$ *including the columns with indices in* $\mathcal{Z}_n \cup (\mathcal{Z}_n+6) \cup (\mathcal{Z}_n+12)$, *denoted by* $\mathbf{F}'(n)$, *should contain at least* 2 *linearly independent left null vectors. However, if we choose the last* 6 *rows in* $\mathbf{F}'(n)$ *uniformly i.i.d. in* $\mathbb{F}_\mathsf{q}$, $\mathbf{F}'(n)$ *whose dimension is* 12×12 *is full rank, and thus does not contain any non-zero left null vector.*

So the main challenge is to design the last 6 *rows in* $\mathbf{F}'$ *such that the following two constraints are satisfied:*

- *(c1) for each* $n \in [\mathsf{N}]$, *the rank of* $\mathbf{F}'(n)$ *is* 10, *such that* $\mathbf{F}'(n)$ *contains* 2 *linearly independent left null vectors; let* $\mathbf{s}^{n,1}$ *and* $\mathbf{s}^{n,2}$ *be these two vectors;*
- *(c2) for any set of workers* $\mathcal{A} \subseteq [\mathsf{K}]$ *where* $|\mathcal{A}| = \mathsf{N_r} = 6$, *by defining* $\mathcal{A}(i)$ *as the* i^{th} *smallest element of* $\mathcal{A}$,

$$\mathbf{S}^{\mathcal{A}} = \begin{bmatrix} \mathbf{s}^{\mathcal{A}(1),1} \\ \mathbf{s}^{\mathcal{A}(1),2} \\ \mathbf{s}^{\mathcal{A}(2),1} \\ \vdots \\ \mathbf{s}^{\mathcal{A}(6),2} \end{bmatrix} \tag{16}$$

with dimension 12×12 *is full rank.*

Note that Condition (c2) guarantees the successful decoding, because if it holds, the master can recover $\mathbf{F}'\mathbf{W}$ *from the answers of any* $\mathsf{N_r}$ *workers.*

By Constraint (c1), the rank of $\mathbf{F}'(n)$ *is* 10, *while the dimension of* $\mathbf{F}'(n)$ *is* 12×12. *Hence, two columns of* $\mathbf{F}'(n)$ *should be linear combinations of the other columns; in other words, we should proceed a rank reduction step on* $\mathbf{F}'(n)$. *It will be explained later in Remark 1 that, if we proceed the rank reduction step individually for each worker, Constraint (c2) cannot be satisfied. Instead, a smart cooperative rank reduction approach is taken across the workers.*

Our approach is based on the following key observation: under the cyclic assignment, each adjacent/neighbouring $\mathsf{N_r} - \mathsf{m} - \mathsf{K_c} = 2$ *workers do not have* $\mathsf{K_c} + 1 = 3$ *common datasets. For example, both workers* 1 *and* 2 *do not have* D_4, D_5, D_6. *We generate one linear equation on the columns of* $\mathbf{F}'$ *with indices in* $\{4,5,6,10,11,12,16,17,18\}$ *(see (18)), we can take one reduction on* $\mathbf{F}'(1)$ *and* $\mathbf{F}'(2)$ *simultaneously. In other words, there exists one vector* $\mathbf{e}_1 = (0,0,0,e_{1,4},e_{1,5},e_{1,6},0,0,0,e_{1,10},e_{1,11},e_{1,12},0,0,0,e_{1,16},e_{1,17},e_{1,18})$ *such that*

$$\mathbf{F}'\mathbf{e}_1^T = 0_{12\times 1}, \tag{18}$$

where $0_{m\times n}$ *represents a matrix of all* 0 *with dimension* $m \times n$. *In addition, by the first two rows of* $\mathbf{F}'$, *it can be seen that* $(e_{1,4}, e_{1,5}, e_{1,6})$ *should be a multiple of* $(1,-2,1)$, *since* $\begin{bmatrix} 1 & 1 & 1 \\ 4 & 5 & 6 \end{bmatrix} [1,-2,1]^T = 0_{2\times 1}$. *Similarly,* $(e_{1,10}, e_{1,11}, e_{1,12})$ *and* $(e_{1,16}, e_{1,17}, e_{1,18})$ *should also be multiples of* $(1,-2,1)$. *By randomly selecting multiple numbers, we let* $(e_{1,4}, e_{1,5}, e_{1,6}) = (2,-4,2)$, $(e_{1,10}, e_{1,11}, e_{1,12}) = (0,0,0)$, *and* $(e_{1,16}, e_{1,17}, e_{1,18}) = (2,-4,2)$.

$$\mathbf{E} = \begin{bmatrix} \mathbf{e}_1 \\ \vdots \\ \mathbf{e}_6 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 2 & -4 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & -4 & 2 \\ 0 & 0 & 0 & 0 & 0 & 0 & 2 & 0 & 0 & 0 & -10 & 8 & 1 & 0 & 0 & 0 & -5 & 4 \\ 8 & -10 & 0 & 0 & 0 & 2 & 4 & 5 & 0 & 0 & 0 & 1 & 8 & -10 & 0 & 0 & 0 & 2 \\ 2 & -4 & 2 & 0 & 0 & 0 & 2 & -4 & 2 & 0 & 0 & 0 & 2 & -4 & 2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -2 & 1 & 0 & 0 & 0 & 2 & -4 & 2 & 0 & 0 \\ 0 & 0 & 1 & -2 & 1 & 0 & 0 & 0 & 1 & -2 & 1 & 0 & 0 & & 2 & -4 & 2 & 0 \end{bmatrix} \tag{17}$$

Similarly, for each $i \in [6]$, the workers in $\{i, Mod(i+1, 6)\}$ do not have $D_{Mod(i-1,6)}, D_{Mod(i-2,6)}, D_{Mod(i-3,6)}$; we generate a vector $\mathbf{e}_i$, where $\mathbf{E} = [\mathbf{e}_1; \dots; \mathbf{e}_6]$ is given in (17) at the top of this page. Note that, each worker $n \in [6]$ is involved in two sets of $\{i, Mod(i+1, 6)\}$ where $i \in [6]$; thus the rank of $\mathbf{F}'(n)$ is reduced by 2, satisfying Constraint (c1).

The next step is to solve the last 6 rows of $\mathbf{F}'$ satisfying

$$\mathbf{F}'\mathbf{E}^T = 0_{12\times 6}. \tag{19}$$

This equation is solvable because $\mathbf{E}^T$ has dimension 18×6 and is full rank; thus the left null space contains $18-6$ linearly independent vectors, which could be exactly the rows of $\mathbf{F}'$.

Finally, as described above, after determining $\mathbf{F}'$, we can then determine $s^{n,1}$ and $s^{n,2}$ for each worker $n \in [6]$. The realizations of $\mathbf{F}'$ and $\mathbf{S}^{[6]}$ are given in [23, Appendix A] respectively. It can be seen that $\mathbf{S}^{[6]}$ is full rank, satisfying Constraint (c2). The general coding scheme will be found in the extended version of this paper [23]

Decoding phase: *Since $\mathbf{S}^{[6]}$ is full rank, by receiving $\mathbf{S}^{[6]}\mathbf{F}'\mathbf{W}$, the master can multiple it by the first 6 rows of the inverse of $\mathbf{S}^{[6]}$ to recover the computation task.* □

Remark 1. *By treating each sub-message which cannot be computed by one worker as an 'interference', in the above example we propose to use the interference alignment strategy to reduce the dimensions of interferences to workers cooperatively. More precisely, we generate the matrix $\mathbf{E}$ in (17) to align interferences, such that 19 holds. We need to generate two equations with the form 18 to align interferences for each worker. If the alignment on the interferences is not perfectly cooperative cross the workers as the proposed scheme (each equation with the form (18) is useful to 2 workers), the number of rows in $\mathbf{E}$ will be strictly larger than 6; then the left null space of $\mathbf{E}^T$ will be strictly less than $18-6=12$, i.e., the number of rows in $\mathbf{F}'$. As a result, $\mathbf{S}^{[6]}$ cannot be full rank.* □

At the end of this paper, we provide some numerical evaluations on the proposed computing scheme. A benchmark scheme is also considered which repeats the computing scheme in [18] K_c times. It can be seen from Fig. 1 and 2 that our proposed scheme has a $(\mathsf{m}+\mathsf{K}_\mathrm{c}-1)$ decrease gain on the communication cost compared to the benchmark scheme, which means with the parameters increase our coding scheme's effect increase. The proposed scheme also coincides with the converse bound under the cyclic assignment in Theorem 1 when K divides N and has a constant gap within 2 when K cannot divide N.

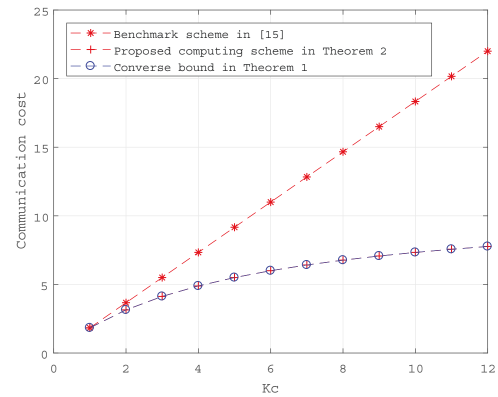

Fig. 1: Communication costs for $\mathsf{K}=\mathsf{N}=12, \mathsf{N}_\mathrm{r}=11, \mathsf{m}=3$

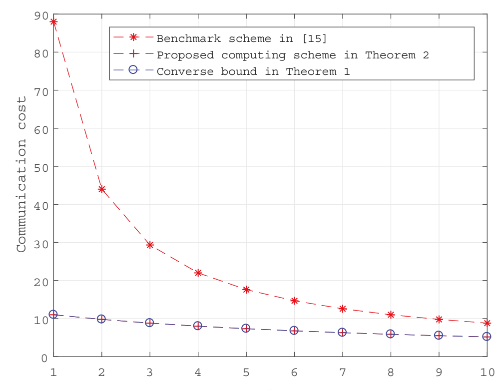

Fig. 2: Communication costs for $\mathsf{K}=\mathsf{N}=12, \mathsf{N}_\mathrm{r}=11, \mathsf{K}_\mathrm{c}=4$

Acknowledgement: The work of W. Huang, K. Wan, and R. C. Qiu was partially funded by the National Natural Science Foundation of China NSFC-12141107. The work of M. Ji was supported in part by National Science Foundation (NSF) CAREER Award 2145835. The work of H. Sun was supported in part by NSF Awards 2007108 and 2045656. The work of G. Caire was partially funded by the ERC Advanced Grant N. 789190, CARENET.

References

[1] B. Rong, "6g: The next horizon: From connected people and things to connected intelligence," *IEEE Wireless Communications*, vol. 28, no. 5, pp. 8–8, 2021.

[2] N. Mital, C. Ling, and D. Gündüz, "Secure distributed matrix computation with discrete fourier transform," *IEEE Transactions on Information Theory*, vol. 68, no. 7, pp. 4666–4680, 2022.

[3] A. E. C. Cloud, "Amazon web services," *Retrieved November*, vol. 9, no. 2011, p. 2011, 2011.

[4] B. Wilder, *Cloud architecture patterns: using microsoft azure*. " O'Reilly Media, Inc.", 2012.

[5] E. Bisong, *Building machine learning and deep learning models on Google cloud platform: A comprehensive guide for beginners*. Apress, 2019.

[6] M. Zaharia, M. Chowdhury, M. J. Franklin, S. Shenker, and I. Stoica, "Spark: Cluster computing with working sets," in *2nd USENIX Workshop on Hot Topics in Cloud Computing (HotCloud 10)*, 2010.

[7] J. Dean and S. Ghemawat, "Mapreduce: simplified data processing on large clusters," *Communications of the ACM*, vol. 51, no. 1, pp. 107–113, 2008.

[8] J. Wang, C. Xu, J. Zhang, and R. Zhong, "Big data analytics for intelligent manufacturing systems: A review," *Journal of Manufacturing Systems*, vol. 62, pp. 738–752, 2022.

[9] Y. Shi, K. Yang, T. Jiang, J. Zhang, and K. B. Letaief, "Communication-efficient edge ai: Algorithms and systems," *IEEE Communications Surveys & Tutorials*, vol. 22, no. 4, pp. 2167–2191, 2020.

[10] J. Wangni, J. Wang, J. Liu, and T. Zhang, "Gradient sparsification for communication-efficient distributed optimization," *Advances in Neural Information Processing Systems*, vol. 31, 2018.

[11] K. Lee, M. Lam, R. Pedarsani, D. Papailiopoulos, and K. Ramchandran, "Speeding up distributed machine learning using codes," *IEEE Transactions on Information Theory*, vol. 64, no. 3, pp. 1514–1529, 2018.

[12] P. Tseng and S. Yun, "A coordinate gradient descent method for nonsmooth separable minimization," *Mathematical Programming*, vol. 117, no. 1, pp. 387–423, 2009.

[13] K. Wan, H. Sun, M. Ji, and G. Caire, "Distributed linearly separable computation," *IEEE Transactions on Information Theory*, vol. 68, no. 2, pp. 1259–1278, 2021.

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

[15] R. Tandon, Q. Lei, A. G. Dimakis, and N. Karampatziakis, "Gradient coding: Avoiding stragglers in distributed learning," in *Proceedings of the 34th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, D. Precup and Y. W. Teh, Eds., vol. 70. PMLR, 06–11 Aug 2017, pp. 3368–3376. [Online]. Available: https://proceedings.mlr.press/v70/tandon17a.html

[16] S. Dutta, V. Cadambe, and P. Grover, "Short-dot: Computing large linear transforms distributedly using coded short dot products," *Advances In Neural Information Processing Systems*, vol. 29, 2016.

[17] T. Akenine-Moller, E. Haines, and N. Hoffman, *Real-time rendering*. AK Peters/crc Press, 2019.

[18] M. Ye and E. Abbe, "Communication-computation efficient gradient coding," in *Proceedings of the 35th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, J. Dy and A. Krause, Eds., vol. 80. PMLR, 10–15 Jul 2018, pp. 5610–5619. [Online]. Available: https://proceedings.mlr.press/v80/ye18a.html

[19] H. Cao, Q. Yan, X. Tang, and G. Han, "Adaptive gradient coding," *IEEE/ACM Transactions on Networking*, vol. 30, no. 2, pp. 717–734, 2021.

[20] J. T. Schwartz, "Fast probabilistic algorithms for verification of polynomial identities," *Journal of the ACM (JACM)*, vol. 27, no. 4, pp. 701–717, 1980.

[21] R. Zippel, "Probabilistic algorithms for sparse polynomials," in *International symposium on symbolic and algebraic manipulation*. Springer, 1979, pp. 216–226.

[22] R. A. Demillo and R. J. Lipton, "A probabilistic remark on algebraic program testing," *Information Processing Letters*, vol. 7, no. 4, pp. 193–195, 1978.

[23] W. Huang, K. Wan, H. Sun, M. Ji, R. C. Qiu, and G. Caire, "Fundamental limits of distributed linearly separable computation under cyclic assignment," 2023.