

Data Shuffling in Wireless Distributed Computing via Low-Rank Optimization

Kai Yang, *Student Member, IEEE*, Yuanming Shi, *Member, IEEE*, and Zhi Ding, *Fellow, IEEE*

Abstract—Intelligent mobile platforms such as smart vehicles and drones have recently become the focus of attention for onboard deployment of machine learning mechanisms to enable low latency decisions with low risk of privacy breach. However, most such machine learning algorithms are both computation- and-memory intensive, which makes it highly difficult to implement the requisite computations on a single device of limited computation, memory, and energy resources. Wireless distributed computing presents new opportunities by pooling the computation and storage resources among devices. For low-latency applications, the key bottleneck lies in the exchange of intermediate results among mobile devices for *data shuffling*. To improve communication efficiency, we propose a co-channel communication model and design transceivers by exploiting the locally computed intermediate values as side information. A low-rank optimization model is proposed to maximize the achieved degrees-of-freedom (DoF) by establishing the interference alignment condition for data shuffling. Unfortunately, existing approaches to approximate the rank function fail to yield satisfactory performance due to the poor structure in the formulated low-rank optimization problem. In this paper, we develop an efficient DC algorithm to solve the presented low-rank optimization problem by proposing a novel DC representation for the rank function. Numerical experiments demonstrate that the proposed DC approach can significantly improve the communication efficiency whereas the achievable DoF almost remains unchanged when the number of mobile devices grows.

Index Terms—Wireless distributed computing, data shuffling, interference alignment, low-rank optimization, difference-of-convex-functions, DC programming, Ky Fan 2 - k norm.

I. INTRODUCTION

The mass use of smart mobile devices and Internet-of-Things (IoT) devices promotes the prosperity of mobile applications, and also poses great opportunities for mobile edge intelligence thanks to large amounts of collected input data from end devices. Machine learning has become a key enabling technology for big data analytics and diverse artificial intelligence applications, including computer vision and natural language processing. Increasingly, more and more machine learning applications are executing real-time and private tasks on mobile devices, such as augmented reality,

smart vehicles, and drones. However, the ultra-low latency requirement [2] for executing intensive computation tasks of mobile edge intelligence applications imposes an unrealistic burden on the computational capability of resource-constrained mobile devices [3] and ranks as one of the key challenges. Given limited resources of computation, storage and energy at mobile devices, a single device often cannot execute the various computation tasks required in learning and artificial intelligence. Wireless distributed computing [4] promises to support computation intensive intelligent tasks execution on end devices by pooling the computation and storage resources of multiple devices.

Storage size is often one of the key limiting factors in a single device when deploying deep learning model [3], [5]. In wireless distributed computing systems for large-scale intelligent tasks, the dataset (e.g., a feature library of objects) is normally too large to be stored in a single mobile device. In popular distributed computing framework such as MapReduce [6], the dataset shall be split and stored across devices in advance, during the *dataset placement phase*. For focal scenarios where each mobile user collects its own input data (e.g., feature vector of an image) and requests the output of its computation task (e.g., inference result of the image), each mobile device shall perform local computation according to locally stored dataset, which is called the *map phase*. Next, in the *shuffle phase*, the computed intermediate values in map phase are exchanged among devices, the output of each mobile device can be constructed with additional local computations (i.e., *reduce phase*). To enable real-time and low-latency applications, inter-device communications for data shuffling in distributed computing system become the main bottleneck.

To reduce the communication load for data shuffling in distributed computing system, many efforts have focused on designing coded shuffling strategies. The authors of [7] exploited the coded multicast opportunities by proposing a coded scheme called “Coded MapReduce” to reduce the communication load for data shuffling in wireline distributed computing framework. In [4], a scalable framework for wireless distributed computing is designed, where mobile devices are connected to a common access point (AP) such that the data shuffling is accomplished through orthogonal uplink transmission and via broadcasting at the rate of weakest user on downlink transmission. In this communication model, a coding scheme is proposed to reduce the *communication load* (i.e., the *number of information bits*) for data shuffling. However, in wireless networks with limited spectral resources and interference, it is also critical to improve the *communication efficiency*

Part of this paper was presented at the IEEE International Conference on Acoustics, Speech, and Signal Processing, Calgary, Alberta, Canada, Apr. 2018 [1].

K. Yang is with the School of Information Science and Technology, ShanghaiTech University, Shanghai, China, and also with the University of Chinese Academy of Sciences, Beijing, China (e-mail: yangkai@shanghaitech.edu.cn).

Y. Shi is with the School of Information Science and Technology, ShanghaiTech University, Shanghai, China (e-mail: shiym@shanghaitech.edu.cn).

Z. Ding is with the Department of Electrical and Computer Engineering, University of California at Davis, Davis, CA 95616 USA (e-mail: zding@ucdavis.edu).

(i.e., *achieved data rates*) for data shuffling. In this paper, we propose a systematic linear coding approach to improve the communication efficiency in the shuffle phase. To improve spectral efficiency, we assume co-channel transmission in both uplink and downlink. By exploiting the locally computed intermediate values in the map phase as side information, we propose to utilize the interference alignment [8] (IA) technique for transceiver design in data shuffling.

By establishing the IA condition for data shuffling, we further develop a low-rank model to maximize the achievable degrees-of-freedom (DoF), i.e., the first-order characterization for the achievable data rate. Low-rank approaches have attracted enormous attention in machine learning, high-dimensional statistics, and recommendation system [9]. Unfortunately, the non-convexity of rank function makes the resulting low-rank optimization problem highly intractable. A growing volume of research focuses on finding tractable approximations for the rank function and on developing efficient algorithms. In particular, nuclear norm relaxation approach is well-known as the convex surrogate of rank function [9]. However, with poorly structured affine constraints in the proposed low-rank optimization model, convex relaxation approach fails to yield satisfactory performance. To further improve the performance of nuclear norm relaxation and enhance low-rankness, the iterative reweighted least square algorithm IRLS- p [10] ($0 \leq p \leq 1$) is proposed by alternating between minimizing weighted Frobenius norm and updating weights. However, such approach still yields unsatisfactory performance under poorly structured affine constraint, which motivates tight and computationally feasible approximations for the rank function. Recently, a DC (difference-of-convex-functions) [11], [12] representation of the rank function has been proposed in [13] with demonstrated effectiveness. Unfortunately, during each iteration of the DC approach, a nuclear norm minimization problem needs to be solved in terms of a semidefinite program and does not scale well to large problem sizes for the data shuffling problem in wireless distributed computing. Motivated by the various issues in the state-of-the-art, we shall propose a novel DC approach which is computation efficient and applicable for wireless distributed computing scenario.

A. Contributions

In this paper, we propose a co-channel communication model for the data shuffling problem in wireless distributed computing system to improve the communication efficiency. Under this model, we adopt linear coding scheme and establish the interference alignment condition for data shuffling. Furthermore, we propose a low-rank optimization model for transceiver design to support efficient algorithms design. To optimize the transceivers with the proposed low-rank model, we propose a novel DC representation for rank function. Specifically, we observe that if the rank of a matrix is k , its Ky Fan $2-k$ norm should be equal to its Frobenius norm. By alternatively increasing rank and minimizing the difference between the square of Frobenius norm and the square of Ky Fan $2-k$ norm, we develop a novel DC approach for the

presented low-rank optimization problem. The Frobenius norm allows us to further derive the closed-form solution for each iteration. During each iteration only a subspace projection needs to be computed.

The major contributions of this work are summarized as follows:

- 1) We propose a co-channel communication model for the data shuffling problem in wireless distributed computing. We adopt linear coding scheme in this work, and establish the interference alignment condition for transceiver design. A low-rank model is then developed to maximize the achievable DoF satisfying interference alignment conditions.
- 2) To improve communication efficiency, we develop a novel computationally efficient DC algorithm for the low-rank optimization problem. This is achieved by proposing a novel DC representation for rank function. The proposed DC algorithm converges to critical points from arbitrary initial points.
- 3) Numerical experiments demonstrate that with the proposed communication model and DC algorithm, data shuffling in wireless distributed computing can be accomplished with high communication efficiency. The proposed DC algorithm significantly outperforms the nuclear norm relaxation approach and the IRLS algorithm. Furthermore, the communication efficiency is scalable to the number of mobile devices.

This work proposes a systematic framework for efficient data shuffling in wireless distributed computing.

B. Organization and Notation

The rest of this work is organized as follows. Section II describes the system model of wireless distributed computing, including the computation model and the proposed communication model. Section III provides the interference alignment conditions for data shuffling as well as the formulated low-rank model. Section IV introduces our proposed DC approach. We conduct numerical experiments and illustrate the performance of the proposed algorithm and other state-of-art algorithms in Section V before concluding this work in Section VI.

We use $[N]$ to denote the set $\{1, \dots, N\}$ for some positive integer N . \otimes is the Kronecker product operator. The cardinality of a set \mathcal{F} is denoted by $|\mathcal{F}|$. $\det(\cdot)$ denotes the determinant of a matrix.

II. SYSTEM MODEL

In this section, we shall introduce the computation model of wireless distributed computing system, followed by proposing a co-channel transmission communication model for data shuffling.

A. Computation Model

Consider the wireless distributed computing system consisting of K mobile users, where mobile users exchange information over a common wirelessly connected access point (AP) as shown in Fig. 1. Suppose each mobile user is equipped

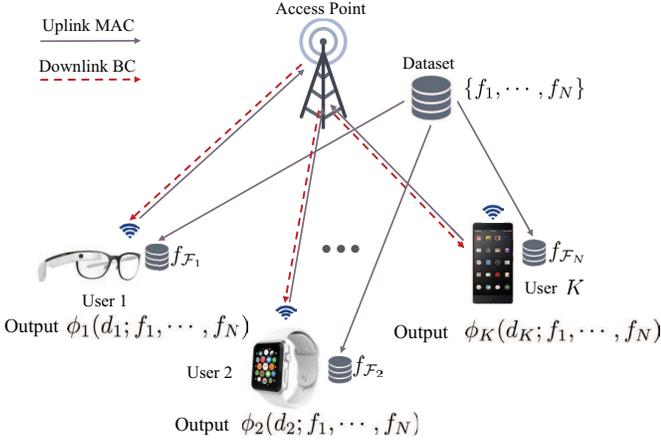


Fig. 1: Wireless distributed computing system.

with L antennas and the AP uses M antennas. The dataset in the system is assumed to be evenly split to N files f_1, \dots, f_N , each with F bits. Each mobile user k aims to obtain the output of computation task $\phi_k(d_k; f_1, \dots, f_N)$ with the input d_k . For example in object recognition, the dataset is a feature library of various objects. Given the feature vector of an image as input, each mobile user requires the inference result of the image. In practice, the storage size of mobile users is often limited [3] and the entire dataset cannot be stored directly at the user end. Therefore, we assume that the local memory size of each mobile user is only μF bits ($\mu < N$), while the whole dataset can be distributively stored across K mobile users (i.e., $\mu K \geq N$). Let $\mathcal{F}_k \subseteq [N]$ be the index set of files stored at user k . Then we have $|\mathcal{F}_k| \leq \mu$ and $\cup_{k \in [K]} \mathcal{F}_k = [N]$. We thus use $f_{\mathcal{F}_k} = \{f_n : n \in \mathcal{F}_k\}$ to denote the set of locally stored files at the k -th mobile user.

In this work, popular distributed computing framework such as MapReduce [6] and Spark is adopted to accomplish all computation tasks, where each computation task ϕ_k is assumed to be decomposed as [4]

$$\phi_k(d_k; f_1, \dots, f_N) = h_k(g_{k,1}(d_k; f_1), \dots, g_{k,N}(d_k; f_N)). \quad (1)$$

In the focused distributed computing architecture, *Map* function $g_{k,n}(d_k; f_n)$ is computed by the k -th mobile user according to file f_n , whose output is the intermediate value $w_{k,n}$ with E bits. The *Reduce* function h_k maps all intermediate values $w_{k,1}, \dots, w_{k,N}$ into the output of computation task ϕ_k . We assume that intermediate values are small enough to be stored at each mobile user while collecting all inputs d_k 's has negligible communication overhead. As shown in Fig. 2, all computation tasks hence can be accomplished via the following four phases:

- **Dataset Placement Phase:** In this phase, the file placement strategy \mathcal{F}_k shall be determined, and files are delivered to the corresponding mobile users in advance to execute Map Phase.
- **Map Phase:** In this phase, intermediate values $w_{k,n}$ are computed locally with map functions $g_{k,n}$ for all $k \in [K]$ and $n \in \mathcal{F}_k$ based on the files $f_{\mathcal{F}_k}$ in the local memory of mobile user k .

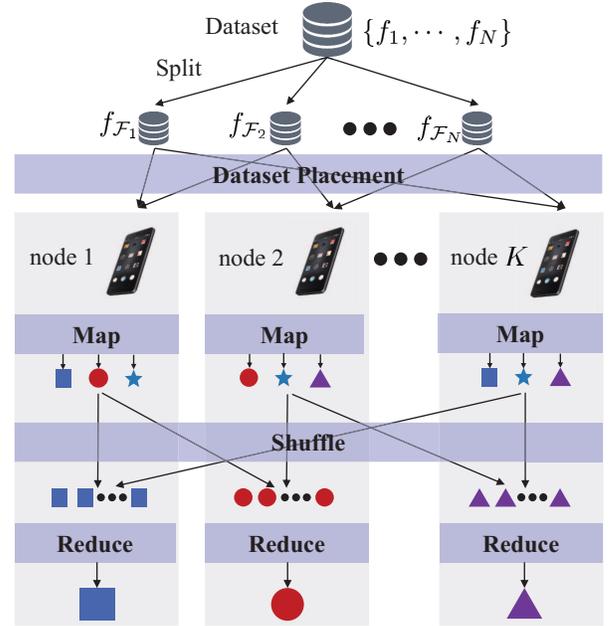


Fig. 2: Distributed computing model.

- **Shuffle Phase:** The output of computation task ϕ_k for mobile user k relies on the intermediate values $\{w_{k,n} : n \notin \mathcal{F}_k\}$ that can only be computed by other mobile users in the Map phase. Therefore, mobile users shall exchange intermediate values wirelessly with each other in this phase.
- **Reduce Phase:** By mapping all required intermediate values into the output value, i.e., $\phi_k(d_k; f_1, \dots, f_N) = h_k(w_{k,1}, \dots, w_{k,N})$, mobile users construct the output of each computation task ϕ_k .

With limited radio resources, data shuffling across mobile devices becomes the significant bottleneck for scaling up wireless distributed computing.

B. Communication Model

In wireless distributed computing systems, communication often becomes the key bottleneck [4] [14] to accomplish the computation tasks. In this paper, we aim to improve the communication efficiency for the Shuffle Phase given the dataset placement. We shall propose a co-channel transmission framework to efficiently exchange the intermediate values for the data shuffling by modeling this problem as a side information aided message delivery problem. Specifically, the set of all intermediate values $\{w_{1,1}, \dots, w_{1,N}, \dots, w_{K,N}\}$ is treated as a library of independent messages $\{W_1, \dots, W_T\}$ with $T = KN$, i.e., the intermediate value $w_{k,n}$ is represented by message $W_{(k-1)N+n}$. Let $\mathcal{T}_k \subseteq [T]$ be the index set of intermediate values available at mobile user k , i.e., $\mathcal{T}_k = \{(j-1)N+n : j \in [K], n \in \mathcal{F}_k\}$. Likewise, let $\mathcal{R}_k \subseteq [T]$ be the index set of intermediate values required by mobile user k where $\mathcal{R}_k = \{(k-1)N+n : n \notin \mathcal{F}_k\}$. Note that $\cup_{k \in [K]} \mathcal{T}_k = [T]$, $\mathcal{T}_k \cap \mathcal{R}_k = \emptyset$ due to the structure of MapReduce-like distributed computing framework. With these notations, the data shuffling in Shuffle Phase is modeled

as a side information aided message delivery problem. The proposed communication model in Shuffle Phase consists of *uplink multiple access (MAC) stage* and *downlink broadcasting (BC) stage*, as shown in Fig. 1. In uplink MAC stage, the AP collects the mixed signal transmitted by all mobile users, and forwards it to each mobile user in downlink BC stage.

Let the aggregated signal transmitted by mobile user k over r channel uses be

$$\mathbf{x}_k = [\mathbf{x}_k[i]] = \begin{bmatrix} \mathbf{x}_k[1] \\ \vdots \\ \mathbf{x}_k[L] \end{bmatrix} \in \mathbb{C}^{Lr}, \quad (2)$$

where $\mathbf{x}_k[i] \in \mathbb{C}^r$ corresponds to the i -th antenna. Let $H_k^{\text{up}}[s, i]$ be the channel coefficient between the i -th antenna of mobile user k and the s -th antenna of AP in uplink MAC stage. The received signal $\mathbf{y}[s] \in \mathbb{C}^r$ at the s -th antenna of AP is given by

$$\mathbf{y}[s] = \sum_{k=1}^K \sum_{i=1}^L H_k^{\text{up}}[s, i] \mathbf{x}_k[i] + \mathbf{n}^{\text{up}}[s], \quad (3)$$

where $\mathbf{n}^{\text{up}}[s] \in \mathbb{C}^r$ is the additive isotropic white Gaussian noise. Here, we consider a quasi-static fading channel model in which channel coefficients remain unchanged over r channel uses. By denoting

$$\mathbf{y} = \begin{bmatrix} \mathbf{y}[1] \\ \vdots \\ \mathbf{y}[M] \end{bmatrix} \in \mathbb{C}^{Mr}, \quad \mathbf{n}^{\text{up}} = \begin{bmatrix} \mathbf{n}^{\text{up}}[1] \\ \vdots \\ \mathbf{n}^{\text{up}}[M] \end{bmatrix} \in \mathbb{C}^{Mr}, \quad (4)$$

$$\mathbf{H}_k^{\text{up}} = \begin{bmatrix} H_k^{\text{up}}[1, 1] & \cdots & H_k^{\text{up}}[1, L] \\ \vdots & \ddots & \vdots \\ H_k^{\text{up}}[M, 1] & \cdots & H_k^{\text{up}}[M, L] \end{bmatrix} \in \mathbb{C}^{M \times L}, \quad (5)$$

the received signal at AP can be written more compactly as

$$\mathbf{y} = \sum_{k=1}^K (\mathbf{H}_k^{\text{up}} \otimes \mathbf{I}_r) \mathbf{x}_k + \mathbf{n}^{\text{up}}, \quad (6)$$

where \otimes denotes Kronecker product.

In the downlink BC stage, the AP forwards the received signal \mathbf{y} to each mobile user. Similarly, the received signal $\mathbf{z}_k \in \mathbb{C}^{Lr}$ by the k -th mobile user is given by

$$\mathbf{z}_k = (\mathbf{H}_k^{\text{down}} \otimes \mathbf{I}_r) \mathbf{y} + \mathbf{n}_k^{\text{down}}, \quad (7)$$

where the channel coefficient matrix $\mathbf{H}_k^{\text{down}}$ in downlink BC stage and the downlink additive isotropic white Gaussian noise $\mathbf{n}_k^{\text{down}}$ are given as

$$\mathbf{H}_k^{\text{down}} = \begin{bmatrix} H_k^{\text{down}}[1, 1] & \cdots & H_k^{\text{down}}[1, M] \\ \vdots & \ddots & \vdots \\ H_k^{\text{down}}[L, 1] & \cdots & H_k^{\text{down}}[L, M] \end{bmatrix} \in \mathbb{C}^{L \times M}, \quad (8)$$

$$\mathbf{n}_k^{\text{down}} = \begin{bmatrix} \mathbf{n}_k^{\text{down}}[1] \\ \vdots \\ \mathbf{n}_k^{\text{down}}[L] \end{bmatrix} \in \mathbb{C}^{Lr}. \quad (9)$$

Therefore, the overall input-output relationship from all mobile users to mobile user k through both the uplink MAC stage and downlink BC stage can be represented as

$$\mathbf{z}_k = \sum_{i=1}^K (\mathbf{H}_k^{\text{down}} \otimes \mathbf{I}_r) (\mathbf{H}_i^{\text{up}} \otimes \mathbf{I}_r) \mathbf{x}_i \quad (10)$$

$$+ (\mathbf{H}_k^{\text{down}} \otimes \mathbf{I}_r) \mathbf{n}^{\text{up}} + \mathbf{n}_k^{\text{down}}$$

$$= \sum_{i=1}^K (\mathbf{H}_{ki} \otimes \mathbf{I}_r) \mathbf{x}_i + \mathbf{n}_k, \quad (11)$$

where $\mathbf{H}_{ki} = \mathbf{H}_k^{\text{down}} \mathbf{H}_i^{\text{up}}$ denotes the equivalent channel state matrix and $\mathbf{n}_k = (\mathbf{H}_k^{\text{down}} \otimes \mathbf{I}_r) \mathbf{n}^{\text{up}} + \mathbf{n}_k^{\text{down}}$ is the effective additive noise.

C. Achievable Data Rates and DoF

Let $R_k(W_l)$ be the achievable data rate of the required message W_l for mobile user k . Then there exists certain coding scheme such that the rate of message W_l is $R_k(W_l)$ while the error probability of decoding W_l for mobile user k can be made arbitrarily small as the length of codewords approaches infinity [15].

As a first-order characterization of channel capacity, degree-of-freedom (DoF) analysis and optimization are widely applied in interference channels [8], [16], [17]. The optimal DoF is also characterized in [8] for the fully connected K user interference channel. Let $\text{SNR}_{k,l}$ be the signal-to-noise-ratio (SNR) therein, followed by the definition of degree-of-freedom [8]

$$\text{DoF}_{k,l} \triangleq \limsup_{\text{SNR}_{k,l} \rightarrow \infty} \frac{R_k(W_l)}{\log(\text{SNR}_{k,l})}. \quad (12)$$

Achievable DoF allocation set is denoted by $\{\text{DoF}_{k,l} : k \in [K], l \notin \mathcal{F}_k\}$ and symmetric DoF (denoted by DoF_{sym}) is defined as the largest achievable DoF for all k, l . That is, the DoF allocation

$$\{\text{DoF}_{k,l} = \text{DoF}_{\text{sym}} : k \in [K], l \notin \mathcal{F}_k\} \quad (13)$$

is achievable. In this paper, we choose DoF as the performance metric for alleviating the interferences in data shuffling. Without loss of generality, we shall maximize the achievable symmetric DoF for the data shuffling in wireless distributed computing, though it can be readily extended to general asymmetric cases.

III. INTERFERENCE ALIGNMENT CONDITIONS AND LOW-RANK FRAMEWORK FOR DATA SHUFFLING

In this section, we shall establish the interference alignment conditions for data shuffling in wireless distributed computing, before developing a low-rank optimization framework for the achievable DoF maximization in linear transceiver design.

A. Interference Alignment Conditions

Linear coding schemes for transceiver design have found applications in interference alignment [8] and index coding [18] owing to its low-complexity and optimality in terms of DoFs. Therefore, we focus on linear coding scheme in this

work. Let $\mathbf{s}_j \in \mathbb{C}^d$ be the representative vector for message W_j with d datastreams such that each datastream carries one DoF. Then the transmitted signal of user k is

$$\mathbf{x}_k = \sum_{j \in \mathcal{T}_k} \mathbf{V}_{kj} \mathbf{s}_j, \quad (14)$$

where \mathbf{V}_{kj} is the precoding matrix of mobile user k for message j and formed by

$$\mathbf{V}_{kj} = \begin{bmatrix} \mathbf{V}_{kj}[1] \\ \vdots \\ \mathbf{V}_{kj}[L] \end{bmatrix} \in \mathbb{C}^{rL \times d}, \quad (15)$$

in which $\mathbf{V}_{kj}[i] \in \mathbb{C}^{r \times d}$ corresponds to the i -th antenna of mobile user k over r channel uses. Likewise, let $\mathbf{U}_{kl} = [\mathbf{U}_{kl}[1] \cdots \mathbf{U}_{kl}[L]] \in \mathbb{C}^{d \times Lr}$ be the decoding matrix for each message W_l with $l \in \mathcal{R}_k$. We then decode message W_l from

$$\tilde{\mathbf{z}}_{kl} = \mathbf{U}_{kl} \mathbf{z}_k = \mathbf{U}_{kl} \sum_{i=1}^K (\mathbf{H}_{ki} \otimes \mathbf{I}_r) \sum_{j \in \mathcal{T}_i} \mathbf{V}_{ij} \mathbf{s}_j + \tilde{\mathbf{n}}_{kl}, \quad (16)$$

where $\tilde{\mathbf{n}}_{kl} = \mathbf{U}_{kl} \mathbf{n}_k$. We observe that $\tilde{\mathbf{z}}_{kl}$ contains the linear combination of the entire message set, which can be decomposed into three parts: the desired message, interferences, and locally available messages, i.e.,

$$\begin{aligned} \tilde{\mathbf{z}}_{kl} = & \mathcal{I}_1(\underbrace{\mathbf{s}_l}_{\text{desired message}}) + \mathcal{I}_2(\underbrace{\{\mathbf{s}_j : j \in \mathcal{T}_k\}}_{\text{locally available messages}}) \\ & + \mathcal{I}_3(\underbrace{\{\mathbf{s}_j : j \notin \mathcal{T}_k \cup \{l\}\}}_{\text{interferences}}) + \tilde{\mathbf{n}}_{kl}. \end{aligned} \quad (17)$$

Specifically, linear operators $\mathcal{I}_1, \mathcal{I}_2, \mathcal{I}_3$ are given by

$$\mathcal{I}_1(\mathbf{s}_l) = \sum_{i:l \in \mathcal{T}_i} \mathbf{U}_{kl} (\mathbf{H}_{ki} \otimes \mathbf{I}_r) \mathbf{V}_{il} \mathbf{s}_l,$$

$$\mathcal{I}_2(\{\mathbf{s}_j : j \in \mathcal{T}_k\}) = \sum_{j \in \mathcal{T}_k} \sum_{i:j \in \mathcal{T}_i} \mathbf{U}_{kl} (\mathbf{H}_{ki} \otimes \mathbf{I}_r) \mathbf{V}_{ij} \mathbf{s}_j,$$

$$\mathcal{I}_3(\{\mathbf{s}_j : j \notin \mathcal{T}_k \cup \{l\}\}) = \sum_{j \notin \mathcal{T}_k \cup \{l\}} \sum_{i:j \in \mathcal{T}_i} \mathbf{U}_{kl} (\mathbf{H}_{ki} \otimes \mathbf{I}_r) \mathbf{V}_{ij} \mathbf{s}_j.$$

Interference alignment [8] turns out to be a powerful tool to handle the mutual interference among users. The basic idea is to make signals resolvable at intended receivers while aligning and cancelling signals at unintended receivers. To eliminate interferences which is the key limit factor for achieving high data rates, we establish the following interference alignment conditions

$$\det \left(\sum_{i:l \in \mathcal{T}_i} \mathbf{U}_{kl} (\mathbf{H}_{ki} \otimes \mathbf{I}_r) \mathbf{V}_{il} \right) \neq 0, \quad (18)$$

$$\sum_{i:j \in \mathcal{T}_i} \mathbf{U}_{kl} (\mathbf{H}_{ki} \otimes \mathbf{I}_r) \mathbf{V}_{ij} = \mathbf{0}, \quad j \notin \mathcal{T}_k \cup \{l\}, \quad (19)$$

where $l \in \mathcal{R}_k, k \in [K]$. By designing transceivers to satisfy interference alignment conditions (18) and (19), message W_l can be decoded from signal $\tilde{\mathbf{s}}_l = \mathcal{I}_1^{-1}(\tilde{\mathbf{z}}_{kl} - \mathcal{I}_2(\{\mathbf{s}_j : j \in \mathcal{T}_k\}))$ for all $l \in \mathcal{R}_k, k \in [K]$.

If conditions (18) and (19) are met, we can obtain interference-free channels for transmitting d -dimensional messages over r channel uses. The achievable DoF $_{k,l}$ is thus given

by d/r . Hence the symmetric DoF in the wireless distributed computing system is given by

$$\text{DoF}_{\text{sym}} = d/r. \quad (20)$$

Consequently, achievable symmetric DoF can be maximized by finding the minimum channel use r subject to (18) and (19).

B. Low-Rank Optimization Approach

In this subsection, we develop a low-rank model to establish the interference alignment conditions (18) and (19) for data shuffling in wireless distributed computing. Note that

$$\mathbf{U}_{kl} (\mathbf{H}_{ki} \otimes \mathbf{I}_r) \mathbf{V}_{ij} = \sum_{m=1}^L \sum_{n=1}^L H_{ki}[m, n] \mathbf{U}_{kl}[m] \mathbf{V}_{ij}[n], \quad (21)$$

where $H_{ki}[m, n]$ is the (m, n) -th entry of matrix \mathbf{H}_{ki} . Define a set of matrices

$$\mathbf{X}_{k,l,i,j} = [\mathbf{X}_{k,l,i,j}[m, n]] = [\mathbf{U}_{kl}[m] \mathbf{V}_{ij}[n]] \quad (22)$$

$$= \begin{bmatrix} \mathbf{U}_{kl}[1] \mathbf{V}_{ij}[1] & \cdots & \mathbf{U}_{kl}[1] \mathbf{V}_{ij}[L] \\ \vdots & \ddots & \vdots \\ \mathbf{U}_{kl}[L] \mathbf{V}_{ij}[1] & \cdots & \mathbf{U}_{kl}[L] \mathbf{V}_{ij}[L] \end{bmatrix} \quad (23)$$

$$= \begin{bmatrix} \mathbf{U}_{kl}[1] \\ \vdots \\ \mathbf{U}_{kl}[L] \end{bmatrix} [\mathbf{V}_{ij}[1] \cdots \mathbf{V}_{ij}[L]] \quad (24)$$

$$= \tilde{\mathbf{U}}_{kl} \tilde{\mathbf{V}}_{ij}, \quad (25)$$

where $\tilde{\mathbf{U}}_{kl} \in \mathbb{C}^{Ld \times r}$ and $\tilde{\mathbf{V}}_{ij} \in \mathbb{C}^{r \times Ld}$. We further denote

$$\mathbf{X} = [\mathbf{X}_{k,l,i,j}] \quad (26)$$

$$= \begin{bmatrix} \mathbf{X}_{1,1,1,1} & \cdots & \mathbf{X}_{1,1,1,T} & \cdots & \mathbf{X}_{1,1,K,T} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ \mathbf{X}_{1,T,1,1} & \cdots & \mathbf{X}_{1,T,1,T} & \cdots & \mathbf{X}_{1,T,K,T} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ \mathbf{X}_{K,T,1,1} & \cdots & \mathbf{X}_{K,T,1,T} & \cdots & \mathbf{X}_{K,T,K,T} \end{bmatrix} \quad (27)$$

$$= \begin{bmatrix} \tilde{\mathbf{U}}_{11} \\ \vdots \\ \tilde{\mathbf{U}}_{1T} \\ \vdots \\ \tilde{\mathbf{U}}_{KT} \end{bmatrix} [\tilde{\mathbf{V}}_{11} \cdots \tilde{\mathbf{V}}_{1T} \cdots \tilde{\mathbf{V}}_{KT}] \quad (28)$$

$$= \tilde{\mathbf{U}} \tilde{\mathbf{V}}, \quad (29)$$

where $\tilde{\mathbf{U}} \in \mathbb{C}^{LdKT \times r}$ and $\tilde{\mathbf{V}} \in \mathbb{C}^{r \times LdKT}$. Without loss of generality, to enable efficient algorithms design, we set $\sum_{i:l \in \mathcal{T}_i} \mathbf{U}_{kl} (\mathbf{H}_{ki} \otimes \mathbf{I}_r) \mathbf{V}_{il} = \mathbf{I}$ in (18). Then the interference alignment conditions (18) and (19) can be rewritten as

$$\sum_{i:l \in \mathcal{T}_i} \sum_{m=1}^L \sum_{n=1}^L H_{ki}[m, n] \mathbf{X}_{k,l,i,l}[m, n] = \mathbf{I}, \quad (30)$$

$$\sum_{i:j \in \mathcal{T}_i} \sum_{m=1}^L \sum_{n=1}^L H_{ki}[m, n] \mathbf{X}_{k,l,i,j}[m, n] = \mathbf{0}, \quad j \notin \mathcal{T}_k \cup \{l\}, \quad (31)$$

which can be characterized by $\mathcal{A}(\mathbf{X}) = \mathbf{b}$ with the linear operator $\mathcal{A} : \mathbb{C}^{D \times D} \mapsto \mathbb{C}^S$ as a function of $\{\mathbf{H}_{ki}\}$. Note that the rank of matrix \mathbf{X} is equal to the number of channel uses r since $\mathbf{X} = \tilde{\mathbf{U}}\tilde{\mathbf{V}}$, i.e.,

$$\text{rank}(\mathbf{X}) = r. \quad (32)$$

We hence propose the following low-rank optimization approach to maximize the achievable symmetric DoF

$$\begin{aligned} \mathcal{P} : & \text{minimize}_{\mathbf{X} \in \mathbb{C}^{D \times D}} \text{rank}(\mathbf{X}) \\ & \text{subject to } \mathcal{A}(\mathbf{X}) = \mathbf{b}, \end{aligned} \quad (33)$$

where $D = LdKT$. However, problem \mathcal{P} is computationally hard due to the non-convexity of the rank function.

C. Problem Analysis

Low-rank optimization approach has recently caught enormous attentions particularly in machine learning, high-dimensional statistics, and recommendation systems [9]. Unfortunately, low-rank optimization problems are generally intractable due to the non-convex rank function. Therefore, many efforts focused on finding tractable representation for the rank function, based on which a number of algorithms are developed.

1) *Nuclear Norm Relaxation*: Nuclear norm [9] has demonstrated its effectiveness as the convex surrogate for the rank function, yielding the following nuclear norm minimization problem

$$\begin{aligned} & \text{minimize}_{\mathbf{X}} \|\mathbf{X}\|_* \\ & \text{subject to } \mathcal{A}(\mathbf{X}) = \mathbf{b}. \end{aligned} \quad (34)$$

The nuclear norm $\|\mathbf{X}\|_*$ is equal to the sum of the singular values of \mathbf{X} . It is the convex hull of the collection of atomic unit-norm rank-one matrices, and is thus the tightest convex relaxation of the rank function. Its equivalent semidefinite programming (SDP) form

$$\begin{aligned} & \text{minimize}_{\mathbf{X}, \mathbf{W}_1, \mathbf{W}_2} \text{Tr}(\mathbf{W}_1) + \text{Tr}(\mathbf{W}_2) \\ & \text{subject to } \mathcal{A}(\mathbf{X}) = \mathbf{b}, \\ & \begin{bmatrix} \mathbf{W}_1 & \mathbf{X} \\ \mathbf{X}^H & \mathbf{W}_2 \end{bmatrix} \succeq \mathbf{0} \end{aligned} \quad (35)$$

can be solved by the interior point method with high precision at a low iteration count. However, this second-order algorithm has high computational complexity with computational cost $\mathcal{O}((S + D^2)^3)$ at each iteration due to the Newton step [19]. The first-order alternating direction method of multipliers (ADMM) [20], [21] significantly reduces the computational cost to $\mathcal{O}(SD^2 + D^3)$ for each iteration (please refer to IV-D for more details). It converges within $\mathcal{O}(1/\epsilon)$ iterations given the precision $\epsilon > 0$.

However, the nuclear norm minimization approach yields unsatisfactory performance due to the poor structure of the affine constraint in problem \mathcal{P} . For example, in the scenario of two users with $K = N = 2, \mu = d = L = M = 1$, each mobile user stores distinct files locally, and requires the

intermediate values computed by the other one. In this case, problem \mathcal{P} is

$$\begin{aligned} & \text{minimize}_{\mathbf{X}} \text{rank}(\mathbf{X}) \\ & \text{subject to } \mathbf{X} = \begin{bmatrix} * & * & * & * & * & * & * & * \\ * & * & * & * & * & * & * & * \\ * & * & * & * & * & * & \frac{1}{H_{12}} & 0 \\ * & * & * & * & * & * & * & * \\ * & * & * & * & * & * & * & * \\ 0 & \frac{1}{H_{21}} & * & * & * & * & * & * \\ * & * & * & * & * & * & * & * \\ * & * & * & * & * & * & * & * \end{bmatrix}, \end{aligned} \quad (36)$$

where the value of $*$ is unconstrained. In this case, the nuclear norm approach always returns full rank solution while the optimal rank is 1. Furthermore, the numerical results provided in Section V shall demonstrate that the convex relaxation approach yields poor performance on average.

2) *Schatten- p Norm Approximation and Iterative Reweighted Least Squares Minimization*: To provide better approximation for the rank function, Schatten- p norm ($0 \leq p \leq 1$) of a matrix has been studied in [10]. Specifically, the Schatten- p norm of matrix $\mathbf{X} \in \mathbb{C}^{D \times D}$ is defined as

$$\|\mathbf{X}\|_p = \left(\sum_{i=1}^D \sigma_i^p(\mathbf{X}) \right)^{1/p}. \quad (37)$$

Since it is nonconvex for $p < 1$, an iterative reweighted least squares algorithm (IRLS- p) is proposed to alternatively minimize weighted Frobenius norm and update weights \mathbf{W} based on the observation that

$$\|\mathbf{X}\|_p^p = \text{Tr}((\mathbf{X}^H \mathbf{X})^{\frac{p}{2}-1} \mathbf{X}^H \mathbf{X}) \quad (38)$$

holds for non-singular matrix \mathbf{X} . In the t -th iteration, \mathbf{X} and weight matrix \mathbf{W} can be updated as follows

$$\mathbf{X}^{[t]} = \underset{\mathbf{X}}{\text{argmin}} \{ \text{Tr}(\mathbf{W}^{[k-1]} \mathbf{X}^H \mathbf{X}) : \mathcal{A}(\mathbf{X}) = \mathbf{b} \} \quad (39)$$

$$\mathbf{W}^{[t]} = (\mathbf{X}^{[t]H} \mathbf{X}^{[t]} + \gamma^{[k]} \mathbf{I})^{\frac{p}{2}-1}, \quad (40)$$

where $\gamma^{[k]} \in \mathbb{R}$ is a regularization parameter to ensure that $\mathbf{W}^{[t]}$ is well-defined and $\{\gamma^{[k]}\}$ is a non-increasing sequence. However, its performance still falls short when applied to problem \mathcal{P} given the poorly structured affine constraint. In this work, we shall propose a novel difference-of-convex-functions (DC) algorithm to achieve considerable performance improvements by rewriting the rank function as a DC function.

IV. DC APPROACH FOR LOW-RANK OPTIMIZATION

This section develops a DC algorithm for the low-rank optimization problem in data shuffling. This is achieved by proposing a novel DC representation for the rank function, and developing an efficient DC algorithm based on the proposed DC representation.

A. DC Approach

A DC representation of the rank function has recently been proposed in [13], followed by a DC algorithm to solve problem \mathcal{P} . We will first introduce the definition of Ky Fan norm.

Definition 1. Ky Fan k -norm [22]: The Ky Fan k -norm of a matrix \mathbf{X} is a convex function of matrix \mathbf{X} and given by the sum of its largest- k singular values, i.e.,

$$\|\mathbf{X}\|_k = \sum_{i=1}^k \sigma_i(\mathbf{X}), \quad (41)$$

where $\sigma_i(\mathbf{X})$ is the i -th largest singular value of \mathbf{X} .

Based on Definition 1, if a matrix is low-rank (rank r), its Ky Fan r -norm equals its nuclear norm. Then a DC representation for the rank function can be obtained. For any matrix $\mathbf{X} \in \mathbb{C}^{m \times n}$, the following equation holds [13]:

$$\text{rank}(\mathbf{X}) = \min\{k : \|\mathbf{X}\|_* - \|\mathbf{X}\|_k = 0, k \leq \min\{m, n\}\}. \quad (42)$$

Therefore, by representing the rank function with Ky Fan k -norm, problem \mathcal{P} can be solved by finding the minimum k such that the optimal objective value is zero in the following optimization problem:

$$\begin{aligned} & \underset{\mathbf{X} \in \mathbb{C}^{D \times D}}{\text{minimize}} && \|\mathbf{X}\|_* - \|\mathbf{X}\|_k \\ & \text{subject to} && \mathcal{A}(\mathbf{X}) = \mathbf{b}, \end{aligned} \quad (43)$$

where the objective is the difference of two convex functions $\|\mathbf{X}\|_*$ and $\|\mathbf{X}\|_k$. Due to the nonconvex DC objective function, the majorization-minimization (MM) algorithm [11], [12] can be adopted to iteratively solve a convex subproblem by linearizing $\|\mathbf{X}\|_k$ as $\text{Tr}(\partial\|\mathbf{X}_t\|_k^H \mathbf{X})$, i.e., by solving

$$\begin{aligned} & \underset{\mathbf{X} \in \mathbb{C}^{D \times D}}{\text{minimize}} && \|\mathbf{X}\|_* - \text{Tr}(\partial\|\mathbf{X}_t\|_k^H \mathbf{X}) \\ & \text{subject to} && \mathcal{A}(\mathbf{X}) = \mathbf{b} \end{aligned} \quad (44)$$

in the $(t+1)$ -th iteration. Here \mathbf{X}_t is the solution to (44) in the t -th iteration. $\partial\|\mathbf{X}_t\|_k$ [22] denotes the subdifferential of $\|\mathbf{X}\|_k$ at \mathbf{X}_t and can be chosen as

$$\partial\|\mathbf{X}_t\|_k = \{\mathbf{U} \text{diag}(\mathbf{q}) \mathbf{V}^H, \mathbf{q} = [\underbrace{1, \dots, 1}_k, \underbrace{0, \dots, 0}_{D-k}]\}, \quad (45)$$

where $\mathbf{X}_t = \mathbf{U} \Sigma \mathbf{V}^H$ is the singular value decomposition (SVD) of \mathbf{X}_t .

Unfortunately, the main drawback of this DC approach is that in each iteration a nuclear norm minimization problem (44) should be solved. The computational cost of nuclear norm minimization problem is $\mathcal{O}(\frac{1}{\epsilon}(SD^2 + D^3))$ even with first-order ADMM algorithm for precision ϵ , which is computationally costly and not amenable to the data shuffling problem in this paper. Efficient algorithm should be proposed especially for the wireless distributed computing scenarios with large number of mobile users. Next, we shall propose a novel computationally efficient DC approach for solving problem \mathcal{P} , for which we propose a novel DC representation for the rank function.

B. A Novel DC Representation for Rank Function

We observe that the nuclear norm function in the objective function of problem (43) leads to cumbersome computations. To overcome the drawback, we propose a novel DC representation of the rank function. We first introduce:

Definition 2. For any integer $1 \leq k \leq \min\{m, n\}$, the Ky Fan 2- k norm [23] of matrix $\mathbf{X} \in \mathbb{C}^{m \times n}$ is defined as the ℓ_2 -norm of the subvector formed by the largest- k singular values of \mathbf{X} . That is,

$$\|\mathbf{X}\|_{k,2} = \left(\sum_{i=1}^k \sigma_i^2(\mathbf{X}) \right)^{1/2}, \quad (46)$$

where $\sigma_i(\mathbf{X})$ is the i -th largest singular value of matrix \mathbf{X} .

The Ky Fan 2- k norm is a unitarily invariant norm, and can be computed via the following SDP problem [23]

$$\begin{aligned} \|\mathbf{X}\|_{k,2}^2 &= \underset{z, \mathbf{U}}{\text{minimize}} && kz + \text{Tr}(\mathbf{U}) \\ & \text{subject to} && z\mathbf{I} + \mathbf{U} \succeq \mathbf{X}^H \mathbf{X}, \\ & && \mathbf{U} \succeq \mathbf{0}. \end{aligned} \quad (47)$$

Note that $\text{rank}(\mathbf{X}) = r$ means that the $\min\{m, n\} - r$ smallest singular values of matrix $\mathbf{X} \in \mathbb{C}^{m \times n}$ are zeros. Based on this fact, we have the following proposition:

Proposition 1. For a matrix $\mathbf{X} \in \mathbb{C}^{m \times n}$, we have

$$\text{rank}(\mathbf{X}) \leq k \Leftrightarrow \|\mathbf{X}\|_F = \|\mathbf{X}\|_{k,2}. \quad (48)$$

Futhermore,

$$\text{rank}(\mathbf{X}) = \min\{k : \|\mathbf{X}\|_F^2 - \|\mathbf{X}\|_{k,2}^2 = 0, k \leq \min\{m, n\}\}. \quad (49)$$

Proof. Given $\text{rank}(\mathbf{X}) \leq k$, we have $\sigma_i(\mathbf{X}) = 0 \forall i > k$. It follows that $\|\mathbf{X}\|_F = \|\mathbf{X}\|_{k,2}$. Conversely, we can deduce $\sigma_i(\mathbf{X}) = 0 \forall i > k$ from $\|\mathbf{X}\|_F = \|\mathbf{X}\|_{k,2}$. Thus, the rank of matrix \mathbf{X} is no more than k .

Let the rank of matrix \mathbf{X} be r . Then $\sigma_i(\mathbf{X}) = 0 \forall i > r$ and $\sigma_i(\mathbf{X}) > 0 \forall i \leq r$. Since $\|\mathbf{X}\|_F = \|\mathbf{X}\|_{k,2}$ if and only if $\text{rank}(\mathbf{X}) \leq k$, the minimum k for $\|\mathbf{X}\|_F^2 - \|\mathbf{X}\|_{k,2}^2 = 0$ will be exactly r . Conversely, $r = \min\{k : \|\mathbf{X}\|_F^2 - \|\mathbf{X}\|_{k,2}^2 = 0\}$ we deduce that $\sigma_i(\mathbf{X}) = 0 \forall i > r$ and $\sigma_i(\mathbf{X}) > 0 \forall i \leq r$. Then $\text{rank}(\mathbf{X}) = r$. \square

C. Efficient DC Algorithm for Problem \mathcal{P}

With the proposed novel DC representation of rank function, the minimum rank r can be found by sequentially solving

$$\begin{aligned} \mathcal{P}_{\text{DC}} : & \underset{\mathbf{X} \in \mathbb{C}^{D \times D}}{\text{minimize}} && \|\mathbf{X}\|_F^2 - \|\mathbf{X}\|_{k,2}^2 \\ & \text{subject to} && \mathcal{A}(\mathbf{X}) = \mathbf{b} \end{aligned} \quad (50)$$

and incrementing k from 1 to $\min\{m, n\}$, until the objective value of problem \mathcal{P}_{DC} achieves zero. Problem \mathcal{P}_{DC} is a DC programming problem since its objective function is the difference of two convex functions.

To develop the simplified form of DC algorithm [11], we equivalently rewrite problem \mathcal{P}_{DC} as

$$\underset{\mathbf{X} \in \mathbb{C}^{m \times n}}{\text{minimize}} \|\mathbf{X}\|_F^2 + I_{\{\mathcal{A}(\mathbf{X})=\mathbf{b}\}}(\mathbf{X}) - \|\mathbf{X}\|_{k,2}^2 \quad (51)$$

where the indicator function I is given by

$$I_{(\mathcal{A}(\mathbf{X})=\mathbf{b})}(\mathbf{X}) = \begin{cases} 0, & \mathcal{A}(\mathbf{X}) = \mathbf{b} \\ +\infty, & \text{otherwise} \end{cases}. \quad (52)$$

To deal with the complex domain, we employ Wirtinger's calculus [24]. Let $g(\mathbf{X}) = \|\mathbf{X}\|_F^2 + I_{(\mathcal{A}(\mathbf{X})=\mathbf{b})}(\mathbf{X})$, $h(\mathbf{X}) = \|\mathbf{X}\|_{k,2}^2$. Since $\{\mathbf{X} : \mathcal{A}(\mathbf{X}) = \mathbf{b}\}$ is an affine subspace, function g and function h are both convex. We denote

$$\alpha = \inf_{\mathbf{X} \in \mathcal{X}} f(\mathbf{X}) = g(\mathbf{X}) - h(\mathbf{X}) \quad (53)$$

where $\mathcal{X} = \mathbb{C}^{m \times n}$. According to the Fenchel's duality [25], its dual problem is given by

$$\alpha = \inf_{\mathbf{Y} \in \mathcal{Y}} h^*(\mathbf{Y}) - g^*(\mathbf{Y}). \quad (54)$$

Here $h^*(\mathbf{Y})$ and $g^*(\mathbf{Y})$ are the conjugate functions of g and h respectively. The conjugate function is defined by

$$g^*(\mathbf{Y}) = \sup_{\mathbf{X} \in \mathcal{X}} \langle \mathbf{X}, \mathbf{Y} \rangle - g(\mathbf{X}), \quad (55)$$

where the inner product is defined as $\langle \mathbf{X}, \mathbf{Y} \rangle = \text{Tr}(\mathbf{X}^H \mathbf{Y})$ based on [24].

Simplified DC algorithm aims to update both the primal and dual variables via successive convex approximation. Specific iterations for solving problem \mathcal{P}_{DC} are given by

$$\mathbf{Y}^{[t]} = \arg \inf_{\mathbf{Y} \in \mathcal{Y}} h^*(\mathbf{Y}) - [g^*(\mathbf{Y}^{[t-1]}) + \langle \mathbf{Y} - \mathbf{Y}^{[t-1]}, \mathbf{X}^{[t]} \rangle], \quad (56)$$

$$\mathbf{X}^{[t+1]} = \arg \inf_{\mathbf{X} \in \mathcal{X}} g(\mathbf{X}) - [h(\mathbf{X}^{[t]}) + \langle \mathbf{X} - \mathbf{X}^{[t]}, \mathbf{Y}^{[t]} \rangle]. \quad (57)$$

Using the Fenchel biconjugation theorem [25], equation (56) can be summarized as

$$\mathbf{Y}^{[t]} \in \partial h(\mathbf{X}^{[t]}). \quad (58)$$

Therefore, we propose to solve problem \mathcal{P}_{DC} by updating the primal and dual variables $\mathbf{X}^{[t+1]}$, $\mathbf{Y}^{[t]}$ via

$$\mathbf{Y}^{[t]} \in \partial \|\mathbf{X}^{[t]}\|_{k,2}^2 \quad (59)$$

$$\mathbf{X}^{[t+1]} = \arg \inf_{\mathbf{X} \in \mathcal{X}} \{\|\mathbf{X}\|_F^2 - \langle \mathbf{X}, \mathbf{Y}^{[t]} \rangle : \mathcal{A}(\mathbf{X}) = \mathbf{b}\}. \quad (60)$$

Proposition 2. One subgradient of $\|\mathbf{X}\|_{k,2}^2$ is given by

$$\partial \|\mathbf{X}\|_{k,2}^2 := 2\mathbf{U}\mathbf{\Sigma}_k\mathbf{V}^H, \quad (61)$$

where $\mathbf{X} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^H$ is the singular value decomposition (SVD) of matrix $\mathbf{X} \in \mathbb{C}^{D \times D}$ and $\mathbf{\Sigma}_k$ keeps the largest k diagonal elements of the matrix $\mathbf{\Sigma}$.

Proof. First we note that the Ky Fan 2- k norm of matrix \mathbf{X} is orthogonally invariant. This can be obtained from the orthogonal invariance of singular values, and

$$\|\mathbf{X}\|_{k,2}^2 = \|\boldsymbol{\sigma}(\mathbf{X})\|_{k,2}^2 = \sum_{i=1}^k \sigma_i^2(\mathbf{X}). \quad (62)$$

Here $\boldsymbol{\sigma} = [\sigma_i(\mathbf{X})] \in \mathbb{R}^D$ denotes the vector composed by all singular values of matrix \mathbf{X} . $\|\boldsymbol{\sigma}(\mathbf{X})\|_{k,2}$ denotes the Ky

Fan 2- k norm of vector $\boldsymbol{\sigma}(\mathbf{X})$. The subgradient of $\|\boldsymbol{\sigma}(\mathbf{X})\|_{k,2}^2$ with respect to $\boldsymbol{\sigma}(\mathbf{X})$ is given by

$$\mathbf{c} \in \mathbb{R}^D : c_i = \begin{cases} 2\sigma_i(\mathbf{X}), & i \leq k \\ 0, & i > k \end{cases}. \quad (63)$$

According to the subdifferential of orthogonally invariant norm [26], we obtain

$$\{\mathbf{U}\text{diag}(\mathbf{d})\mathbf{V}^H : \mathbf{X} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^H, \mathbf{d} \in \partial \|\boldsymbol{\sigma}(\mathbf{X})\|_{k,2}\} \subseteq \partial \|\mathbf{X}\|_{k,2}. \quad (64)$$

It then follows that

$$2\mathbf{U}\mathbf{\Sigma}_k\mathbf{V}^H \in \partial \|\mathbf{X}\|_{k,2}^2, \quad (65)$$

where $\mathbf{\Sigma}_k$ is given by

$$(i, j)\text{-th entry of } \mathbf{\Sigma}_k := \begin{cases} \sigma_i(\mathbf{X}), & i = j, i \leq k \\ 0, & \text{otherwise} \end{cases}. \quad (66)$$

□

Note that each iteration of (59) and (60) for the proposed DC algorithm can be computed much more efficiently than solving the nuclear norm minimization problem (44) since (60) is a simple quadratic programming (QP) problem with closed form solutions. Specifically, according to (59) and (60), $\mathbf{X}^{[t+1]}$ can be rewritten as the solution to the following quadratic program:

$$\begin{aligned} & \text{minimize}_{\mathbf{X} \in \mathbb{C}^{D \times D}} \|\mathbf{X} - \frac{1}{2}\partial \|\mathbf{X}^{[t]}\|_{k,2}^2\|_F^2 \\ & \text{subject to } \mathcal{A}(\mathbf{X}) = \mathbf{b}. \end{aligned} \quad (67)$$

The solution to this least square problem with affine constraint is the orthogonal projection onto the affine subspace, whose closed-form is given by

$$\mathbf{X}^{[t+1]} = (\mathbf{I} - \mathcal{A}^+ \mathcal{A}) \left(\frac{1}{2} \partial \|\mathbf{X}^{[t]}\|_{k,2}^2 \right) + \mathcal{A}^+(\mathbf{b}), \quad (68)$$

where $\mathcal{A}^+ = \mathcal{A}^H(\mathcal{A}\mathcal{A}^H)^{-1}$. Therefore, the overall procedure of our proposed DC algorithm is shown in Algorithm 1.

Algorithm 1: Proposed DC Approach for problem \mathcal{P}

Input: \mathcal{A}, \mathbf{b} .

for $r = 1, \dots, \min\{m, n\}$ **do**

Initialize: $\mathbf{X}_r^{[0]} \in \mathbb{C}_*^{m \times n}$

while not converge do

$\mathbf{X}_r^{[t+1]} = (\mathbf{I} - \mathcal{A}^+ \mathcal{A}) \left(\frac{1}{2} \partial \|\mathbf{X}_r^{[t]}\|_{k,2}^2 \right) + \mathcal{A}^+(\mathbf{b})$

end

if $\text{rank}(\mathbf{X}_r) \leq r$ **then**

return \mathbf{X}_r

end

end

Output: \mathbf{X}_r and $\text{rank}(\mathbf{X}_r)$.

D. Computational Complexity and Convergence Analysis

The proposed DC algorithm involves computing a series of equation (68) multiple times for fixed rank r . Since both \mathcal{A} and \mathcal{A}^+ can be computed and stored in advance, in each iteration the computational overhead comes from matrix vector multiplication and subgradient evaluation. Since the dimension of \mathcal{A} is $\mathbb{C}^{D \times D} \mapsto \mathbb{C}^S$, the complexity of matrix vector multiplication is $\mathcal{O}(SD^2)$. Computing the subgradient by following (61) is dominated by the SVD with computational complexity $\mathcal{O}(D^3)$. Therefore, the computational overhead of the proposed DC algorithm for each iteration is $\mathcal{O}(SD^2 + D^3)$. However, the first-order algorithm ADMM [21] needs to solve a sequence of semidefinite cone projection problem via SVD for solving the nuclear norm minimization problem (44), which yields computational cost $\mathcal{O}(\frac{1}{\epsilon}(SD^2 + D^3))$ with ϵ as the solution accuracy. Therefore, our proposed DC algorithm is much more computationally efficient with closed form solution for solving the DC program (50), instead of solving a nuclear norm minimization problem for solving the DC program (43) using the algorithm in [13]. The complexity of the iterations (39) and (40) for the IRLS- p algorithm using projected gradient descent method [10] is $\mathcal{O}((SD^2 + D^3) \log \frac{1}{\epsilon})$.

The proposed DC algorithm can be implemented very efficiently due to the sparse structure of operator \mathcal{A} . Therefore, the overhead of matrix vector multiplication is often small especially when L and d are much smaller compared with the number of involved mobile users. Specifically, the sparsity level of the linear operator \mathcal{A} is given as

$$\sum_{k=1}^K \sum_{l \in \mathcal{R}_k} \sum_{j \neq \mathcal{T}_k} |\{i : j \in \mathcal{T}_i\}| L^2 d^2. \quad (69)$$

For example, for a single-antenna wireless distributed computing system with 5 mobile users and 10 files in the dataset, if each mobile user stores 6 files in its local storage unit and messages are delivered with single datastream, $D = 250$ and the sparsity level of \mathcal{A} is only 920.

The convergence of the proposed DC algorithm for solving problem \mathcal{P}_{DC} is given by the following proposition.

Proposition 3. Given rank parameter k , the proposed Algorithm 1 for solving problem \mathcal{P}_{DC} converges to critical points from arbitrary initial points.

Proof. Please refer to Appendix A for details. \square

V. NUMERICAL RESULTS

In this section, we describe numerical experiments to compare the performance of the proposed DC algorithm (Algorithm 1) with the following benchmarks:

- **Nuclear norm relaxation:** To evaluate the performance of the nuclear norm relaxation approach (35), we implement the interior point method introduced in Section III-C1 with CVX [27] toolbox.
- **Iterative reweighted least squares (IRLS):** In [10], smoothed Schatten- p norm approximation for the rank function is adopted. To solve this nonconvex problem, the iterative reweighted least squares algorithm is proposed

as presented in Section III-C2. p is chosen as 0.5 through cross validation in this section.

In all simulations, we consider the symmetric case where all mobile users and the AP are equipped with $L = M$ antennas. The maximum achievable symmetric DoF (20) is chosen as the performance metric. The channel coefficients are randomly drawn from independent and identically distributed complex Gaussian distribution, i.e., $\mathbf{H}_{ki} \sim \mathcal{CN}(\mathbf{0}, \mathbf{I})$. For each algorithm, the rank is determined by the number of singular values above 10^{-5} . Given r , iterations for the proposed DC algorithm will be terminated when the $(r + 1)$ -th singular value is less than 10^{-5} , i.e., $\sigma_{r+1}(\mathbf{X}) < 10^{-5}$.

A. Achievable DoF over Local Storage Size

Consider a wireless distributed computing framework with 5 single-antenna mobile users and a single-antenna AP. Each mobile user stores 5 to 9 files locally while the full dataset consists of 10 files. We shall evaluate the maximum achievable symmetric DoF that each algorithm can obtain with the assumption that each message is a single datastream. We run each algorithm 100 replications to evaluate the relationship between DoF and the local storage size.

From Fig. 3, we observe that the achievable symmetric DoF has visible growth when more files are stored at each mobile devices for all algorithms. Clearly, this is because more cooperation is enabled and fewer intermediate values need to be exchanged when each mobile user can access more files of the whole dataset. The proposed DC algorithm outperforms both the IRLS algorithm and nuclear norm relaxation. The result of this experiment demonstrates that the proposed DC representation for the rank function has advantages over the Schatten- p norm approximation approach, while the nuclear norm relaxation is inferior to the other two approaches.

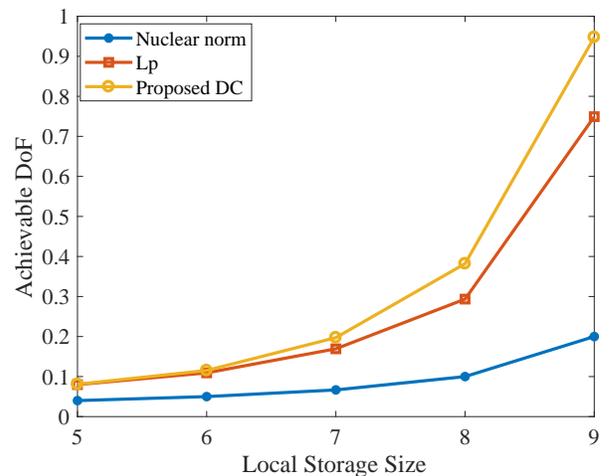


Fig. 3: The maximum achievable symmetric DoF over local storage size μ of each mobile user.

B. Achievable DoF over the Number of Antennas

We consider a wireless distributed computing framework with 8 mobile users and an AP. Each mobile user stores 1 out

of 4 files in its local memory. We assume that each mobile users and the AP are equipped with the same number of antennas. We used different number of antennas to evaluate the multiplex gain of the focused wireless distributed computing system. Each point is averaged 100 times and the result is shown in Fig. 4.

We can see that achievable symmetric DoF grows linearly with the number of antennas for the proposed DC algorithm and IRLS algorithm. However, the achievable DoF by the nuclear norm relaxation algorithm remains constant despite the growing number of antennas due to the poor structure of our problem. This test demonstrates that the proposed transceiver design framework achieves linear gain by increasing the number of antennas for the proposed DC algorithm. It also shows the intrinsic defects of nuclear norm relaxation approach for the data shuffling problem. The proposed DC approach is superior to the IRLS algorithm and the nuclear relaxation approach for data shuffling.

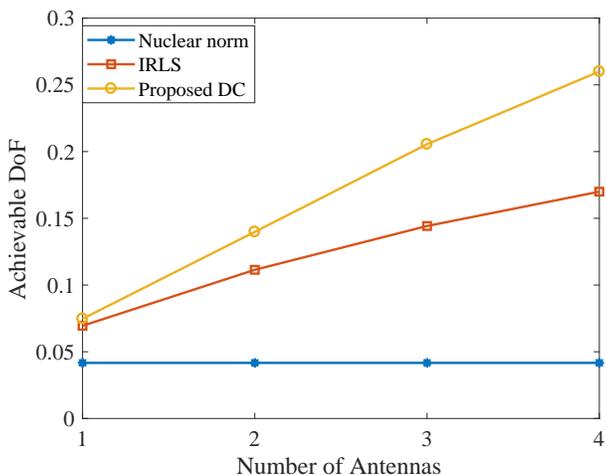


Fig. 4: The maximum achievable symmetric DoF over the number of antennas when the mobile users and the AP are equipped with same number of antennas.

C. Achievable DoF over the Number of Mobile Users

As pointed in [4], the limited communication bandwidth may become the bottleneck since the computation tasks increase linearly with network size. Therefore, the scalability becomes critical for a wireless distributed computing framework. In this test, we shall evaluate the achievable DoF by increasing the number of mobile users. Consider a single-antenna wireless distributed computing system where the dataset can be separated to 5 files, and each mobile user can only store up to 2 files in its local storage. We consider the uniform placement case when each mobile user stores $\mu = 2$ files and each file is stored by $\mu K/N = 2K/5$ mobile users. Consider the single datastream case of $d = 1$. The achievable symmetric DoFs of different algorithms averaged over 100 trials are shown in Fig. 5. The achievable DoFs of the proposed DC algorithms remain nearly unchanged as the network size grows, which demonstrates its scalability. On the contrary, there is a marked

decline of the achievable DoFs for IRLS algorithm and nuclear norm relaxation algorithm. Although more requested messages are involved in the system when the number of users grows, opportunities of collaboration for mobile users also increase since each file is stored at more mobile users. Our proposed algorithm can harness the benefits of such collaboration while other algorithms fail. However, it still remains an interesting but challenging problem to prove the scalability theoretically for the proposed DC algorithm.

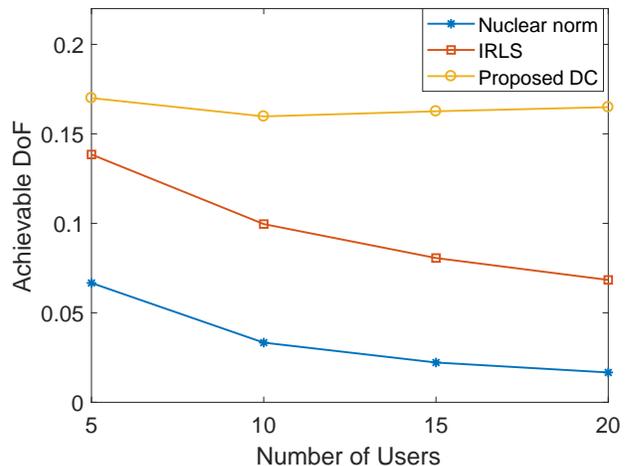


Fig. 5: The achievable DoF with different algorithms over the number of mobile users.

In summary, the proposed DC algorithm has the capability of achieving higher DoF over benchmark approaches by exploiting the special structure of the data shuffling problem. Furthermore, the achievable DoF of the proposed DC algorithm almost remains unchanged when the number of mobile users increases.

VI. CONCLUSION

In this paper we proposed a novel low-rank optimization to improve the communication efficiency for wireless distributed computing. We focus on the data-shuffle phase of the distributed computing and establish a novel interference alignment condition for data shuffling. We proposed a novel DC representation for the rank function based on Ky Fan $2-k$ norm, and then developed an efficient DC algorithm for the focused low-rank optimization problem, by deriving the closed-form solution for each iteration of the proposed DC algorithm. Numerical results demonstrated that the proposed DC approach can achieve higher DoF than the nuclear norm relaxation approach and IRLS algorithm. Furthermore, in uniform placement scenario, the achievable DoF nearly remains unchanged though more mobile users are involved.

For the proposed data shuffling strategy for wireless distributed computing, there still exist some open problems. Possible future directions are listed as follows:

- Although we have shown that the proposed low-rank approach is scalable with the growth of mobile users, it is particularly interesting to prove the scalability theoretically.

- We have shown that the proposed DC algorithm converges globally, but establishing the convergence rate can be considered in future works.
- It would also be interesting to consider the transceiver design with finite SNR scenarios for the proposed communication model for data shuffling in wireless distributed computing systems.

APPENDIX A
PROOFS OF PROPOSITION 3: CONVERGENCE OF
ALGORITHM 1

Since $\mathbf{Y}^{[t]} \in \partial h(\mathbf{X}^{[t]})$, we have

$$h(\mathbf{X}^{[t+1]}) \geq h(\mathbf{X}^{[t]}) + \langle \mathbf{X}^{[t+1]} - \mathbf{X}^{[t]}, \mathbf{Y}^{[t]} \rangle. \quad (70)$$

Hence, it follows

$$(g-h)(\mathbf{X}^{[t+1]}) \leq g(\mathbf{X}^{[t+1]}) - \langle \mathbf{X}^{[t+1]} - \mathbf{X}^{[t]}, \mathbf{Y}^{[t]} \rangle - h(\mathbf{X}^{[t]}). \quad (71)$$

Similarly, $\mathbf{X}^{[t+1]} \in \partial g^*(\mathbf{Y}^{[t]})$ implies

$$g(\mathbf{X}^{[t]}) \geq g(\mathbf{X}^{[t+1]}) + \langle \mathbf{X}^{[t]} - \mathbf{X}^{[t+1]}, \mathbf{Y}^{[t]} \rangle + \|\mathbf{X}^{[t+1]} - \mathbf{X}^{[t]}\|_F^2. \quad (72)$$

Thus, we obtain inequality

$$\begin{aligned} g(\mathbf{X}^{[t+1]}) - \langle \mathbf{X}^{[t+1]} - \mathbf{X}^{[t]}, \mathbf{Y}^{[t]} \rangle - h(\mathbf{X}^{[t]}) \\ \leq (g-h)(\mathbf{X}^{[t]}) - \|\mathbf{X}^{[t+1]} - \mathbf{X}^{[t]}\|_F^2. \end{aligned} \quad (73)$$

On the other hand,

$$\mathbf{X}^{[t+1]} \in \partial g^*(\mathbf{Y}^{[t]}) \Leftrightarrow \langle \mathbf{X}^{[t+1]}, \mathbf{Y}^{[t]} \rangle = g(\mathbf{X}^{[t+1]}) + g^*(\mathbf{Y}^{[t]}) \quad (74)$$

$$\mathbf{Y}^{[t]} \in \partial h(\mathbf{X}^{[t]}) \Leftrightarrow \langle \mathbf{X}^{[t]}, \mathbf{Y}^{[t]} \rangle = h(\mathbf{X}^{[t]}) + h^*(\mathbf{Y}^{[t]}). \quad (75)$$

Then it follows

$$\begin{aligned} g(\mathbf{X}^{[t+1]}) - \langle \mathbf{X}^{[t+1]} - \mathbf{X}^{[t]}, \mathbf{Y}^{[t]} \rangle - h(\mathbf{X}^{[t]}) \\ = h^*(\mathbf{Y}^{[t]}) - g^*(\mathbf{Y}^{[t]}). \end{aligned} \quad (76)$$

According to (71) and (73), we obtain that

$$\begin{aligned} (g-h)(\mathbf{X}^{[t+1]}) &\leq h^*(\mathbf{Y}^{[t]}) - g^*(\mathbf{Y}^{[t]}) \\ &\leq (g-h)(\mathbf{X}^{[t]}) - \|\mathbf{X}^{[t+1]} - \mathbf{X}^{[t]}\|_F^2. \end{aligned} \quad (77)$$

Adding that

$$(g-h)(\mathbf{X}) \geq 0, \quad (78)$$

the objective value converges and

$$\lim_{t \rightarrow \infty} \|\mathbf{X}^{[t+1]} - \mathbf{X}^{[t]}\|_F^2 = 0. \quad (79)$$

For every limit point,

$$(g-h)(\mathbf{X}^{[t+1]}) = (g-h)(\mathbf{X}^{[t]}), \quad (80)$$

and

$$\|\mathbf{X}^{[t+1]} - \mathbf{X}^{[t]}\|_F^2 = 0. \quad (81)$$

Therefore, we have

$$(g-h)(\mathbf{X}^{[t+1]}) = h^*(\mathbf{Y}^{[t]}) - g^*(\mathbf{Y}^{[t]}) = (g-h)(\mathbf{X}^{[t]}). \quad (82)$$

From (75) we know that

$$h(\mathbf{X}^{[t+1]}) + h^*(\mathbf{Y}^{[t]}) = g(\mathbf{X}^{[t+1]}) + g^*(\mathbf{Y}^{[t]}) = \langle \mathbf{X}^{[t+1]}, \mathbf{Y}^{[t]} \rangle, \quad (83)$$

i.e.,

$$\mathbf{Y}^{[t]} \in \partial h(\mathbf{X}^{[t+1]}). \quad (84)$$

Then we have $\mathbf{Y}^{[t]} \in \partial g(\mathbf{X}^{[t+1]}) \cap \partial h(\mathbf{X}^{[t+1]})$, which implies that $\mathbf{X}^{[t+1]}$ is a critical point of $g-h$. Therefore, given r Algorithm 1 converges to critical points from arbitrary initial points.

REFERENCES

- [1] K. Yang, Y. Shi, and Z. Ding, "Low-rank optimization for data shuffling in wireless distributed computing," *Proc. IEEE Int. Conf. Acoustics Speech Signal Process. (ICASSP)*, 2018.
- [2] V. Sze, Y. H. Chen, T. J. Yang, and J. S. Emer, "Efficient processing of deep neural networks: A tutorial and survey," *Proc. IEEE*, vol. 105, no. 12, pp. 2295–2329, Dec. 2017.
- [3] S. Han, H. Mao, and W. J. Dally, "Deep compression: Compressing deep neural networks with pruning, trained quantization and Huffman coding," *Int. Conf. Learn. Representations (ICLR)*, 2016.
- [4] S. Li, Q. Yu, M. A. Maddah-Ali, and A. S. Avestimehr, "A scalable framework for wireless distributed computing," *IEEE/ACM Trans. Netw.*, vol. 25, no. 5, pp. 2643–2654, Oct. 2017.
- [5] Y. Cheng, D. Wang, P. Zhou, and T. Zhang, "Model compression and acceleration for deep neural networks: The principles, progress, and challenges," *IEEE Signal Process. Mag.*, vol. 35, no. 1, pp. 126–136, Jan. 2018.
- [6] J. Dean and S. Ghemawat, "MapReduce: simplified data processing on large clusters," *Commun. ACM*, vol. 51, no. 1, pp. 107–113, 2008.
- [7] 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, Jan. 2018.
- [8] V. R. Cadambe and S. A. Jafar, "Interference alignment and degrees of freedom of the k -user interference channel," *IEEE Trans. Inf. Theory*, vol. 54, no. 8, pp. 3425–3441, Aug. 2008.
- [9] M. A. Davenport and J. Romberg, "An overview of low-rank matrix recovery from incomplete observations," *IEEE J. Sel. Topics Signal Process.*, vol. 10, no. 4, pp. 608–622, Jun. 2016.
- [10] K. Mohan and M. Fazel, "Iterative reweighted algorithms for matrix rank minimization," *J. Mach. Learn. Res.*, vol. 13, pp. 3441–3473, Nov. 2012.
- [11] P. D. Tao and L. T. H. An, "Convex analysis approach to DC programming: Theory, algorithms and applications," *Acta Mathematica Vietnamica*, vol. 22, no. 1, pp. 289–355, 1997.
- [12] H. A. Le Thi and T. P. Dinh, "DC programming and DCA: thirty years of developments," *Math. Program.*, pp. 1–64, 2018.
- [13] J.-y. Gotoh, A. Takeda, and K. Tono, "DC formulations and algorithms for sparse optimization problems," *Math. Program.*, vol. 169, no. 1, pp. 141–176, May 2018.
- [14] Y. Lin, S. Han, H. Mao, Y. Wang, and W. J. Dally, "Deep gradient compression: Reducing the communication bandwidth for distributed training," *Int. Conf. Learn. Representations (ICLR)*, 2017.
- [15] T. M. Cover and J. A. Thomas, *Elements of information theory*. John Wiley & Sons, 2012.
- [16] Y. Shi, J. Zhang, and K. B. Letaief, "Low-rank matrix completion for topological interference management by Riemannian pursuit," *IEEE Trans. Wireless Commun.*, vol. 15, no. 7, pp. 4703–4717, Jul. 2016.
- [17] G. Bresler, D. Cartwright, and D. Tse, "Feasibility of interference alignment for the MIMO interference channel," *IEEE Trans. Inf. Theory*, vol. 60, no. 9, pp. 5573–5586, Sept. 2014.
- [18] H. Maleki, V. R. Cadambe, and S. A. Jafar, "Index coding – an interference alignment perspective," *IEEE Trans. Inf. Theory*, vol. 60, no. 9, pp. 5402–5432, Sept. 2014.
- [19] S. Boyd and L. Vandenberghe, *Convex optimization*. Cambridge Univ. Press, 2004.
- [20] Y. Shi, J. Zhang, B. O'Donoghue, and K. B. Letaief, "Large-scale convex optimization for dense wireless cooperative networks," *IEEE Trans. Signal Process.*, vol. 63, no. 18, pp. 4729–4743, Sept. 2015.
- [21] B. O'Donoghue, E. Chu, N. Parikh, and S. Boyd, "Conic optimization via operator splitting and homogeneous self-dual embedding," *J. Optim. Theory Appl.*, vol. 169, no. 3, pp. 1042–1068, Jun 2016.

- [22] G. Watson, "On matrix approximation problems with Ky Fan k norms," *Numerical Algorithms*, vol. 5, no. 5, pp. 263–272, 1993.
- [23] X. V. Doan and S. Vavasis, "Finding the largest low-rank clusters with ky fan 2-k-norm and ℓ_1 -norm," *SIAM J. Optim.*, vol. 26, no. 1, pp. 274–312, 2016.
- [24] P. Bouboulis, K. Slavakis, and S. Theodoridis, "Adaptive learning in complex reproducing kernel Hilbert spaces employing Wirtinger's subgradients," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 23, no. 3, pp. 425–438, Mar. 2012.
- [25] R. T. Rockafellar, *Convex analysis*. Princeton university press, 2015.
- [26] G. A. Watson, "Characterization of the subdifferential of some matrix norms," *Linear algebra and its applications*, vol. 170, pp. 33–45, 1992.
- [27] M. Grant and S. Boyd, "CVX: Matlab software for disciplined convex programming, version 2.1," <http://cvxr.com/cvx>, Mar. 2014.