A New Interference-Alignment Scheme for Wireless MapReduce

Yue Bi*,†, Michèle Wigger†, Yue Wu*

*School of Electronic Information and Electrical Engineering, Shanghai Jiao Tong University, China; wuyue@sjtu.edu.cn
† LTCI, Telecom Paris, IP Paris, 91120 Palaiseau, France; {bi, michele.wigger}@telecom-paris.fr

Abstract—We consider a full-duplex wireless Distributed Computing (DC) system under the MapReduce framework. New upper and lower bounds on the optimal tradeoff between Normalized Delivery Time (NDT) and computation load are presented. The upper bound strictly improves over the previous reported upper bounds and is based on a novel interference alignment (IA) scheme tailored to the interference cancellation capabilities of MapReduce nodes. The lower bound is proved through information-theoretic converse arguments.

Index Terms—Wireless distributed computing, MapReduce, coded computing, interference alignment.

I. INTRODUCTION

Distributed Computing (DC) systems are computer networks that through task-parallelization reduce execution times of complex computing tasks such as data mining or computer vision. MapReduce is a popular such framework and runs in three phases [1], [2]. In the first map phase, nodes calculate intermediate values (IVA) from their associated input files. In the following shuffle phase, nodes exchange these IVAs in a way that each node obtains all IVAs required to compute its assigned output function in the final reduce phase. MapReduce is primarily applied to wired systems where it has been noticed that a significant part of the MapReduce execution time stems from the IVA delivery time during the shuffle phase [2], [3], and can be reduced through smart coding [3]–[8].

MapReduce systems are becoming important building blocks also in wireless scenarios, such as vehicular networks [9] or distributed e-health applications [10], thus creating a need for good wireless MapReduce coding schemes. Similarly to the wired case [4]–[6], delivery time in wireless MapReduce systems can be decreased by sending appropriate linear combinations of the IVAs, from which the receiving nodes can extract their desired IVAs by bootstrapping the IVAs that they can compute from their locally stored input files. Further improvements can however be achieved by exploiting specific wireless communication techniques.

The focus of this paper is on the high Signal-to-Noise Ratio (SNR) regime, and on the following two key metrics:

• Computation load $r$: This describes the average number of nodes to which each file is assigned.
• Normalized Delivery Time (NDT) $\Delta$: This is the wireless shuffle duration normalized by the number of reduce functions and input files and by the transmission time of a single IVA over a point-to-point channel.

We are interested in the minimal NDT for given computation load $r$, which we call the NDT-computation tradeoff.

The NDT of full-duplex interference networks was considered in [11], [12], see [13] for the half-duplex network. In [11], the authors proposed to divide the nodes into groups and let each group store a subset of the files and apply one-shot beamforming and zero-forcing during the shuffle communication. As shown in [11], their scheme is optimal among this class of communication strategies. In our previous work [12], we reduced this NDT by introducing the IA technique to the shuffle communication in [11].

In this paper, we obtain yet a further NDT reduction by considering the map procedure in [4], where each set of $r$ nodes stores a subset of the files, and by proposing a novel IA scheme that is tailored to this file assignment and the interference cancellation capabilities of MapReduce nodes so as to obtain an improved performance compared to standard IA schemes. Our scheme is related to the IA scheme in [14], which considers a similar file assignment, and to our previous work [12], with which it coincides when each node can only store a single file.

The upper bound on the NDT implied by our new IA-scheme improves over the previously proposed bounds in [11], [12] for a system with $K$ nodes whenever the computation load $r$ satisfies $1 < r < \left[\frac{K}{2}\right]$. Our results show that in this regime, beamforming and zero-forcing cannot achieve minimum NDT. On the contrary, through an information-theoretic lower bound on the minimum NDT, we show that for $r \geq \frac{K}{2}$ the zero-forcing and interference cancellation scheme in [11] is optimal also without any restriction on the utilized coding scheme.

Notations: We use standard notation, and also define $[n] \triangleq \{1,2,\ldots,n\}$ and $[A]^n$ as the collection of all the subsets of $A$ with cardinality $n$.

II. WIRELESS MAPREDUCE FRAMEWORK

Consider a distributed computing (DC) system with a fixed number of $K$ nodes labelled $1,\ldots,K$; a large number $N$ of input files $W_1,\ldots,W_N$; and $K$ output functions $\phi_1,\ldots,\phi_K$ mapping the input files to the desired computations.

A MapReduce System decomposes the output functions as:

$$\phi_q(W_1,\ldots,W_N) = v_q(a_{q,1},\ldots,a_{q,N}), \quad q \in [K],$$

(1)

where $v_q$ is called reduce function and $a_{q,p}$ the intermediate value (IVA) calculated from file $W_p$ through a map function

$$a_{q,p} = u_{q,p}(W_p), \quad p \in [N].$$

(2)
IVAs are independent with A i.i.d. bits.

The MapReduce framework has 3 phases:

Map phase: A subset of all input files $\mathcal{M}_k \subseteq [N]$ is assigned to each node $k \in [K]$. Node $k$ computes all IVAs \{$a_{q,p}; p \in \mathcal{M}_k, q \in [K]$\} associated with these input files. Notice that the set \{$\mathcal{M}_k$\}$k \in [K]$ is a design factor.

Shuffle phase: Computation of the $k$-th output function is assigned to the $k$-th node.

The $K$ nodes in the system communicate over $T$ uses of a wireless network in a full-duplex mode to exchange the missing IVAs for the computations of their assigned output functions. So, node $k \in [K]$ produces complex channel inputs of the form

$$X_k \triangleq (X_k(1), \ldots, X_k(T))T = f_k^{(T)}(\{a_{1,p}, \ldots, a_{K,p} | p \in \mathcal{M}_k\}),$$

by means of an encoding function $f_k^{(T)}$ on appropriate domains and so that the inputs satisfy the block-power constraint

$$\frac{1}{T} \sum_{t=1}^{T} E[|X_k(t)|^2] \leq P, \quad k \in [K].$$

Given the full-duplex nature of the network, Node $k$ also observes the complex channel outputs

$$Y_k(t) = \sum_{k' \in [K] \setminus \{k\}} H_{k,k'}(t)X_{k'}(t) + Z_k(t), \quad t \in [T],$$

where the sequences of complex-valued channel coefficients \{$H_{k,k'}(t)$\} and standard circularly symmetric Gaussian noises \{$Z_k(t)$\} are both i.i.d. and independent of each other and of all other channel coefficients and noises.

Based on its outputs $Y_k \triangleq (Y_k(1), \ldots, Y_k(T))T$ and the IVAs \{$a_{q,p}; p \in \mathcal{M}_k, q \in [K]$\} computed during the Map phase, Node $k$ decodes the missing IVAs $\{a_{q,p}; p \notin \mathcal{M}_k\}$ required to compute its assigned output functions $\phi_k$ as:

$$\hat{\phi}_k = g_k^{(T)}(\{a_{1,i}, \ldots, a_{K,i} | i \in \mathcal{M}_k, Y_k\}), \quad p \notin \mathcal{M}_k.$$  

Reduce phase: Each node $k \in [K]$ applies reduce functions $\phi_k(\cdot)$ to the appropriate IVAs calculated during the Map phase or decoded in the Shuffle phase.

The performance of the distributed computing system is measured in terms of its computation load

$$r \triangleq \sum_{k \in [K]} \frac{|\mathcal{M}_k|}{N},$$

and the normalized delivery time (NDT)

$$\Delta \triangleq \lim_{P \to \infty} \lim_{A \to \infty} \frac{T}{A \cdot K \cdot N} \cdot \log P.$$  

We focus on the fundamental NDT-computation tradeoff $\Delta^*_r$, which is defined as the infimum over all $\Delta$ satisfying (8) for some choice of file assignments \{$\mathcal{M}_k$\} and sequence (in $T$) of encoding and decoding functions \{$f_k^{(T)}$\} and \{$g_k^{(T)}$\} in (3) and (6), all depending on $A$ so that the probability of IVA decoding error

$$\Pr\left[\bigcup_{k \in [K]} \bigcup_{p \notin \mathcal{M}_k} \hat{\phi}_k(p) \neq a_{q,p} \right] \to 0 \quad \text{as} \quad A \to \infty.$$

A. Sufficiency of Symmetric File Assignments

Our model exhibits a perfect symmetry between the various nodes in the network because the channels from any Tx-node to any Rx-node are independent and have identical statistics. The optimal NDT-computation tradeoff can therefore be achieved by a symmetric file assignment where any subset of nodes $T \subseteq [K]$ of size $i$ is assigned the same number of files to be stored at all nodes in $T$. In fact, any non-symmetric file assignment can be symmetrized by time-sharing as mentioned in section IV.

B. Relation to the Sum-DoF with r-fold Cooperation

A well-studied property of wireless networks is the Sum Degrees of Freedom (sum-DoF), which describes the maximum throughput of a network. We are specifically interested in the sum-DoF that one can achieve over the wireless network described by (5), when the inputs are subject to the constraints (4) and any set of $r$ nodes $T \subseteq [K]$ has a message $M^T_r$ that it wishes to convey to Node $j$, for any $j \in [K] \setminus T$. Each message $M^T_r$ is uniformly distributed over a set $[2^rR^T_r]$ and a rate-tuple $(R^T_r; T \subseteq [K]^r, j \in [K] \setminus T)$ is called achievable if there exists a sequence of encoding and decoding functions such that the probability of errors tend to $0$ in the asymptotic regime of infinite blocklengths. The sum-DoF is defined as

$$\text{Sum-DoF}(r) \triangleq \sup_{P \to \infty} \lim_{P \to \infty} \frac{\sum_{T \subseteq [K]^r, j \in [K] \setminus T} R^T_r(P)}{\frac{1}{2} \log P},$$

where the supremum is over sequences \{$(R^T_r(P); T \subseteq [K]^r, j \in [K] \setminus T)$\}$P > 0$ so that for each $P > 0$ each tuple $(R^T_r(P); T \subseteq [K]^r, j \in [K] \setminus T)$ is achievable under power $P$.

**Lemma 1.** For any $r \in [K]$:

$$\Delta(r) \geq \left(1 - \frac{r}{K}\right) \frac{1}{\text{Sum-DoF}(r)}.$$  

**Proof:** See the long version [15], but the idea is well known and also used in [11] and [12].

III. NEW IA SCHEME

In view of Lemma 1, we present a scheme achieving a high Sum-DoF($r$) over the wireless network. We illustrate our scheme for $r = 2$ and $K = 4$, which is not optimal in this case but easy to understand.

A. Example 2: $K = 4, r = 2$

In this case our scheme transmits 22 different messages depicted in (12). Here, Message $M^1_{k,T}$ is a message that is known by the set of nodes $T$ and intended to Node $j \notin T$. (Since we consider r-fold cooperation, we have $|T| = 2$.)

Message $M^1_{k,T}$ is only transmitted by a single Node $k \in T$. The remaining nodes in $T \setminus \{k\}$ only exploit their knowledge of $M^1_{k,T}$ to cancel the transmission from their receive signal. Notice that for certain sets $T$ and receive nodes $j \notin T$ our scheme sends two messages to the same node $j$: $M^1_{k_1,T}$ and $M^1_{k_2,T}$ for $k_1 \neq k_2$. These messages $M^1_{k_1,T}$ and $M^1_{k_2,T}$ represent two independent submessages of Message $M^1_T$ as we
defined it in Section II-B. For the sets $\mathcal{T}$ and Nodes $j \notin \mathcal{T}$ for which there exists only a single Message $M_{k,T}^j$, this message is really the message $M_{k}^j$ as we defined it in Section II-B. Since our interest is on the Sum-DoF, distinction between submessages and messages is not relevant.

In our scheme, to Node 1 we send messages
\[ M_{2,1}^1, M_{3,1}^1, M_{2,2}^1, M_{3,2}^1; \]  
(12a)

to Node 2 we send messages
\[ M_{2,1}^2, M_{3,1}^2, M_{2,2}^2, M_{3,2}^2; \]  
(12b)

to Node 3 we send messages
\[ M_{1,1}^3, M_{2,1}^3, M_{3,1}^3, M_{4,1}^3; \]  
(12c)

to Node 4 we send messages
\[ M_{1,2}^4, M_{2,2}^4, M_{3,2}^4, M_{4,2}^4; \]  
(12d)

Node $K = 4$ does not send any message to the first Node 1. We encode each message $M_{k,T}^j$ into a Gaussian codeword $b_{k,T}^j$ and use IA with three precoding matrices $U_{(2,3)}$, $U_{(2,4)}$, and $U_{(3,4)}$. Matrix $U_{(2,3)}$ is used to send codewords
\[ b_{1,1}^2, b_{1,2}^2, b_{2,1}^2, b_{2,2}^2; \]  
(13)

desired signal
\[ b_{1,1}^3, b_{1,2}^3, b_{2,1}^3, b_{2,2}^3; \]  
(14)

matrix $U_{(2,4)}$ for codewords
\[ b_{1,1,4}^2, b_{1,2,4}, b_{2,1,4}^2, b_{2,2,4}^2; \]  
(15)

and matrix $U_{(3,4)}$ for codewords
\[ b_{1,1,3}^4, b_{1,2,3}^4, b_{2,1,3}^4, b_{2,2,3}^4; \]  
(16)

\[ b_{1,2}^1, b_{2,1}^1, b_{2,2}^1, b_{1,1}^1; \]  
(17)

\[ b_{1,1}^4, b_{1,2}^4, b_{2,1}^4, b_{2,2}^4; \]  
(18)

**Remark 1.** The choice of precoding matrices is inspired by [16] [14] where Message $M_{k,T}^j$ is precoded by the matrix $U_{R}$ for $R = T\backslash\{k\} \cup \{j\}$. Since any node in $R$ is either interested in learning Message $M_{k,T}^j$ or it can compute it itself and remove the interference, any node only experiences interference from precoding matrices $U_{R}$ for which $j \notin R$. In our IA scheme, we omit precoding matrices $U_{R}$ for sets $R'$ containing index 1, and instead use the precoding matrix $U_{R}$ also to send
\[ b_{k,R}^j, b_{k,R\cup\{1\}\backslash\{j\}}^j, \quad \forall j, k \in R, j \neq k, \]  
(19)

see the codewords indicated in (14), (16), (18).

We illustrate our assignment of the precoding matrices also in Table I. The entries in column 1 or in rows $\{1, 2\}, \{1, 3\}, \{1, 4\}$ correspond to two submessages $M_{k_1,T}^j$ and $M_{k_2,T}^j$, where $k_1$ and $k_2$ denote the two entries in $T$. For all other entries in Table I not equal to “x”, we have only one message per precoding matrix, see (13), (15), and (17).

During the shuffling phase, Node 1 sends:
\[ X_1 = U_{(2,3)} \left( b_{1,1,3}^2 + b_{1,1,2}^3 \right) \]  
\[ + U_{(2,4)} \left( b_{1,1,4}^2 + b_{1,1,2}^4 \right) \]  
\[ + U_{(3,4)} \left( b_{1,1,4}^3 + b_{1,1,3}^4 \right). \]  
(20)

Nodes 2 and 3 send similar signals based on the assigned codewords. Node 4 sends:
\[ X_4 = U_{(2,3)} \left( b_{1,2,3}^2 + b_{1,2,4}^3 \right) \]  
\[ + U_{(2,4)} b_{1,2,4}^2 + U_{(3,4)} b_{1,2,4}^3. \]  
(21)

Each node subtracts all the interference of the signals that it can compute itself. For example, Node 2 thus constructs:
\[ Y_2' = H_{2,1} U_{(2,3)} b_{1,1,3}^2 + H_{2,1} U_{(2,4)} b_{1,1,4}^2 \]  
\[ + H_{2,3} U_{(2,3)} b_{1,3,3}^2 + H_{2,3} U_{(2,4)} b_{1,3,4}^2 \]  
\[ + H_{2,4} U_{(2,3)} b_{1,4,3}^2 + H_{2,4} U_{(2,4)} b_{1,4,4}^2 \]  
\[ + H_{2,1} U_{(3,4)} b_{1,4,3}^3 + H_{2,3} U_{(3,4)} b_{1,4,3}^4 \]  
\[ + H_{2,1} U_{(3,4)} b_{1,4,4}^3 + H_{2,3} U_{(3,4)} b_{1,4,4}^4 \]  
\[ + H_{2,4} U_{(3,4)} b_{1,4,4}^2 + Z_2. \]  
(22)

**Table I**

| Messages $M_{k,T}^j$ precoded by the three precoding matrices $U_{(2,3)}$, $U_{(2,4)}$, and $U_{(3,4)}$. |
|---|---|---|---|
| $T \backslash j$ | 1 | 2 | 3 |
| {1, 2} | x | x | $U_{(2,3)}$ | $U_{(2,4)}$ |
| {1, 3} | x | $U_{(2,3)}$ | x | $U_{(2,4)}$ |
| {1, 4} | x | $U_{(2,4)}$ | $U_{(3,4)}$ | x |
| (2, 3) | $U_{(2,3)}$ | x | x | $U_{(2,4)}$, $U_{(3,4)}$ |
| (2, 4) | $U_{(2,4)}$ | x | $U_{(2,3)}$, $U_{(3,4)}$ | x |
| (3, 4) | $U_{(3,4)}$, $U_{(2,3)}$, $U_{(2,4)}$ | x | x |
where each exponent-vector $\alphaR = (\alphaR, \HR : \HR \in \H(R)) \in [\eta]^{T}$; $\eta$ is a large number depending on the blocklength $T$ that tends to $\infty$ with $T$; $\XiR$ are i.i.d. random vectors drawn according to a continuous distribution, and

$$\H(2,3) \triangleq \{H_{1,4}, H_{4,1}, H_{4,2}, H_{4,3}\}, \quad \H(2,4) \triangleq \{H_{1,3}, H_{3,1}, H_{3,2}, H_{3,4}\}, \quad \H(3,4) \triangleq \{H_{1,2}, H_{2,1}, H_{2,3}, H_{2,4}\}. \quad (24)$$

Thus, for each set $\mathcal{R} \in [[K]]^{T}$, without causing non-desired interference to nodes in $\mathcal{R}$, we can use the same precoding matrix $U_{\mathcal{R}}$ for all the codewords:

$$\{b_{1,\mathcal{R} \cup \{k\} \setminus \{j\}} \}_{k \in [K] \setminus \mathcal{R}} \quad (32)$$

This idea was already used in the related works [13], [14]. In contrast to these previous works, here we do not introduce the precoding matrices $U_{\mathcal{R}}$ for sets $\mathcal{R}$ containing 1 and instead we use matrix $U_{\mathcal{R}}$, for $1 \notin \mathcal{R}$, also to precode the codewords

$$\{b_{1,\mathcal{R} \cup \{1\} \setminus \{j\}} \}_{j \in \mathcal{R}} \cup \{b_{1,\mathcal{R} \cup \{1\} \setminus \{j\}} \}_{j \notin \mathcal{R}} \quad (33)$$

and

$$\{b_{1,\mathcal{R} \cup \{k\}} \}_{k \in \mathcal{R}}. \quad (34)$$

All non-intended nodes in $\mathcal{R}$ can subtract these interferences from their receive signals because they know the codewords. This trick allows us to reduce the dimension of the interference space and thus improve performance.

Table II illustrates which codesymbols $b_{1,\mathcal{T}}$ are premultiplied by the precoding matrix $U_{\mathcal{T}}$, when $r = 3$ and $K \geq 5$. The entries in rows $T$ containing index 1 correspond to $r = 3$ different codewords $b_{1,\mathcal{T}}$, one for each $k \in T$, see (33). Similarly, the entry in column 1 and row $[2, 3, 4]$ corresponds to the $r$ codewords $b_{1,\mathcal{T}}$, for each $k \in [2, 3, 4]$. Any other entry of the table showing $U_{\mathcal{T}}$ corresponds to a single codeword $b_{1,\mathcal{T}}$, where $k$ is the single element in $T \setminus [2, 3, 4]$. Similar tables can be drawn for all pairs $(k_1, k_2) \in [K]$, where recall that node $k$ does not send any information to Node 1.

**Encoding:** Define the $T$-length vector of channel inputs $X_k \triangleq (X_k(1), \ldots, X_k(T))^T$ for each Node $k$ and set:

$$X_1 = \sum_{\mathcal{R} \in [[K] \setminus \{j\}]^T} \sum_{j \in \mathcal{R}} U_{\mathcal{R}} b_{1,\mathcal{R} \cup \{1\} \setminus \{j\}}. \quad (35)$$

$$X_k = \sum_{\mathcal{R} \in [[K] \setminus \{1\}]^T} \sum_{j \in \mathcal{R}} U_{\mathcal{R}} b_{1,\mathcal{R} \cup \{k\} \setminus \{j\}}$$

$$+ \sum_{\mathcal{R} \in [[K] \setminus \{\{1\}\}]^T} \sum_{j \in \mathcal{R}} U_{\mathcal{R} \cup \{k\} \setminus \{j\}} b_{1,\mathcal{R} \cup \{k\} \setminus \{j\}}. \quad (36)$$
\[
X_K = \sum_{\mathcal{R} \in [K\setminus \{1\}]} \sum_{j \in \mathcal{R}} U_{\mathcal{R}}^j b_{K,R,\cup(k)}^j \]
where we shortly describe matrices \( \{ U_{\mathcal{R}} \} \) \( \mathcal{R} \in [K\setminus \{1\}]^{*} \). 

Decoding: After receiving the respective sequence of \( T \) channel outputs \( Y_j \equiv (Y_{j,1}, \ldots, Y_{j,T}) \), for \( j \in [K] \), each node removes the influence of the codewords corresponding to the messages that it can compute itself. The nodes’ “cleaned” signals can then be written as:

\[
Y_j' = \sum_{\mathcal{R} \in [K\setminus \{1\}]^*} \sum_{k \in [K\setminus \{1\}]^*} H_{j,k} U_{\mathcal{R} \cup \{k\}} \sum_{j \in \mathcal{R}}^1 b_{K,R,\cup(k)}^j + Z_j,
\]

\[
Y_j'' = \sum_{\mathcal{R} \in [K\setminus \{1\}]^*} \sum_{k \in [K\setminus \{1\}]^*} H_{j,k} U_{\mathcal{R} \cup \{k\}} \sum_{j \in \mathcal{R}}^1 b_{K,R,\cup(k)}^j + Z_j,
\]

where for ease of notation we defined for Nodes \( k \in [K\setminus \{1\}] \):

\[
v_{\mathcal{R} \cup \{k\}} = \sum_{j \in \mathcal{R}} b_{K,R,\cup(k)}^j,
\]

for the last Node \( K \), since its signal to Node 1 is absent:

\[
v_{\mathcal{R} \cup \{k\}} = \sum_{j \in \mathcal{R} \cup \{k\}},
\]

Each Node \( j \) zero-forces the non-desired interference terms of its “cleaned” signal and decodes its intended messages.

Choice of IA Matrices \( \{ U_{\mathcal{R}} \} \): Inspired by the IA scheme in [17], we choose each \( T \times n^2 \) precoding matrix \( U_{\mathcal{R}} \) so that its column-span includes all power products (powers 1 to \( n \)) of the channel matrices \( H_{j,k} \) that premultiply \( U_{\mathcal{R}} \) in (38) in the non-desired interference terms. Thus, \( \mathcal{R} \in [K\setminus \{1\}]^* \):

\[
U_{\mathcal{R}} \equiv \prod_{H_{\mathcal{R}}} H_{\mathcal{R},H} \cdot \Xi_{\mathcal{R}}: \forall \alpha_{\mathcal{R}} \in [n]^{\mathcal{R}},
\]

where \( \{ \Xi_{\mathcal{R}} \} \) \( \mathcal{R} \in [K\setminus \{1\}]^* \) are i.i.d. random vectors independent of all channel matrices, noises, and messages.

Performance Analysis: See the long version [15].

IV. NEW BOUNDS ON THE NDT

Define for any integer value \( r \in [K] \):

\[
\Delta_{ub}(r) \begin{cases} 
(1 - \frac{r}{K}) \cdot \frac{r(K-1)+K-r-1}{r(K-r+1)} & \text{if } \ r < K/2 \\
\frac{1}{K} \left( 1 - \frac{r}{K} \right) & \text{if } \ r \geq K/2 
\end{cases}
\]

Also, let

\[
\Delta_{lb}(r) \begin{cases} 
\frac{1}{K} \left( 2 - \frac{3}{K} \right) & \text{if } \ r = 1, \\
\frac{1}{K} \left( 1 - \frac{r}{K} \right) + \max_{r \in [K/2]} \text{lowe}(C_i(r)) & \text{if } \ r \in (1, \frac{K}{2}), \\
\frac{1}{K} \left( 1 - \frac{r}{K} \right) & \text{if } \ r \in \left[ \frac{K}{2}, K \right],
\end{cases}
\]

where for any \( t \in [K/2] \):

\[
C_i(t) = \begin{cases} 
\frac{(r-1)}{(r)} \cdot (K-2t) & \text{if } i \in \{t\}, \\
0 & \text{if } i \in [K\setminus \{t\}] 
\end{cases}
\]

and for any function \( f \), \( \text{lowe}(f(t)) \) denotes the lower convex envelope of \( \{t, f(t)\} \).

Theorem 1. The NDT-computation tradeoff \( \Delta^*(r) \) is upper- and lower-bounded as:

\[
\Delta_{lb}(r) \leq \Delta^*(r) \leq \text{lowe}(\Delta_{ub}(r)).
\]

Proof: For integers \( r \geq K/2 \) achievability of \( \Delta_{ub}(r) \) is proved in [11]. For integers \( r < K/2 \) achievability of \( \Delta_{ub}(r) \) holds by Lemma 1 and the scheme in Section III. Achievability of the lower convex envelope follows by simple time- and memory-sharing strategies. The lower bound can be proved using MAC-type arguments, see our long version [15].

Remark 2. The upper bound is convex and piece-wise constant. The lower bound is piecewise constant with segments spanning the intervals \( [i, i+1] \), for \( i = 2, \ldots, K-1 \). On the interval \( [1, 2] \), the lower bound is constant over smaller sub-intervals only but not over the entire segment.

Corollary 1. For all \( r \geq [K/2] \), the linear interference cancellation scheme in [11] achieves the NDT, which equals

\[
\Delta^*(r) = \left(1 - \frac{r}{K}\right) \cdot \frac{1}{K}.
\]

Proof: For \( r \geq [K/2] \) the upper bound \( \text{lowe}(\Delta_{ub}(r)) \) is equal to the lower bound \( \Delta_{lb}(r) \) because \( C_{[K/2]}(i) = 0 \) for all \( i \geq [K/2] \).

Remark 3. By [11], \( \Delta^*(r) \) in (46) is achieved with beamforming, zero-forcing, and side-information cancellation. By Corollaries 1 and 2, these simple strategies are sufficient to achieve \( \Delta^*(r) \) when \( r \geq [K/2] \) but not when \( r < [K/2] \).
We compare the upper bound in Theorem 1 to the bounds in [11] and [12]. The upper bound in [11] is given as follows:
\[
\Delta^*(r) \leq \Delta_{\text{UB-BF}}(r) \triangleq \text{lowc} \left\{ \left( r, \frac{1-r/K}{\min(K,2r)} \right) : r \in [K] \right\}
\] (47)

The upper bound in [12] has the form:
\[
\Delta^*(r) \leq \Delta_{\text{UB-Groups}}(r) \triangleq \text{lowc} \left\{ \left( K, 0 \right) \cup \left\{ \left( r, \frac{1-r/K}{\text{Sum-DoF}_{\text{Lh}}(r)} \right) : 1 \leq r < K, r|K \right\} \right\},
\] (48)

where
\[
\text{Sum-DoF}_{\text{Lh}}(r) \triangleq \begin{cases} 2r & \text{if } K/r \in \{2,3\}, \\ \frac{K(1-K/r)^{-2}}{2K-3r} & \text{if } K/r \geq 4. \end{cases} \] (49)

Notice that \(\Delta_{\text{UB}}(1) = \Delta_{\text{UB-Groups}}(1)\).

**Corollary 2.** For all \(1 < r < \left[ \frac{K}{2} \right]\):
\[
\Delta^*(r) \leq \text{lowc} \left( \Delta_{\text{UB}}(r) \right) < \Delta_{\text{UB-Groups}}(r),
\] (50)

and for all \(1 \leq r < \left[ \frac{K-1}{2} \right]\):
\[
\Delta^*(r) \leq \text{lowc} \left( \Delta_{\text{UB}}(r) \right) < \Delta_{\text{UB-ZF}}(r).
\] (51)

In Fig. 1 and 2, we plot the mathematical expressions of the bounds in Theorem 1, \(\Delta_{\text{UB-Groups}}(r)\) and \(\Delta_{\text{UB-BF}}(r)\).

![Fig. 1. Bounds on \(\Delta^*(r)\) for \(K = 11\).](image1)

**V. CONCLUSION**

This paper presents an improved upper bound and the first information-theoretic lower bound on the NDT tradeoff of full-duplex wireless MapReduce systems. The upper bound is obtained by zero-forcing and a novel IA scheme that is tailored to the information cancellation capabilities of the nodes in a MapReduce system. As a conclusion of this work, we observe that linear beamforming, zero-forcing, and interference cancellation are optimal when each node can store at least half of the files, but suboptimal otherwise.

**ACKNOWLEDGEMENTS**

We thank P. Ciblat for helpful discussions. This work has been supported by National Key R&D Program of China under Grant No 2020YFB1807504 and National Science Foundation of China Key Project under Grant No 61831007.