Reducing the Average Delay in Gradient Coding

Ming Hui Jovan Lee, Ivan Tjuawinata, Han Mao Kiah
School of Physical and Mathematical Sciences, Nanyang Technological University, Singapore
{mlee031, ivan.tjuawinata, hmkiah}@ntu.edu.sg

***Abstract*—Tandon et al. (2017) introduced a coding theoretic framework to alleviate the problem of stragglers in distributed learning. Following Tandon et al., many authors provided explicit schemes that were able to compute a certain function using $n-s$ replies from n workers in the *worst* case.**

In this work, we focus on reducing the expected delay. To reduce the expected delay, we modify existing schemes so that less than $(n-s)$ replies are sufficient in *most* cases. In particular, we provide a simple modification to existing optimal schemes and demonstrate that with this modification, the expected delay time converges to the fundamental delay. Additionally, for specific parameters, we reduce the number of replies further so that the expected delay time converges faster to the fundamental delay.

I. Introduction

In this information era, we are faced with an abundant amount of data, and this huge amount of data results in a high processing complexity. An example is the training phase of machine learning, where we find a least-cost approximation function for a large dataset. This can be accomplished via *gradient descent*, which entails the iterative computations of a gradient function for the large dataset. Unfortunately, it is often not feasible for one worker, or workstation, to complete the entire computation.

To overcome this issue, distributed computing is used. The simplest way of distributing the workload is to divide the larger dataset into smaller partitions and give one partition to each worker. Each worker then performs the computation on its smaller partition and sends the computation result back to the taskmaster. The desired computation for the entire dataset can then be recovered from *all* replies of the workers. One of the main concerns in this scheme is the existence of stragglers. Stragglers are workers that experience some communication delay. Such delay can be caused by various reasons such as unbalanced load and resource contention [1]. This communication delay can even been shown to last up to five times of the typical behaviour [2]. The severity of delay caused by stragglers becomes more pronounced when the degree of parallelism increases [3].

To alleviate this problem of stragglers, Tandon et al. borrowed ideas from erasure-coding and introduced the notion of *gradient codes* [2]. The crucial idea is this: assign more than one data partition to each worker! When the assignment is judiciously chosen, the taskmaster is able to recover the desired computation from *some* replies.

Optimal schemes in this framework have been provided in recent works [2], [4], [5]. In these works, the authors looked at the worst-case scenario and designed schemes that enables taskmaster to perform its computation whenever it receives a certain number of replies. Furthermore, Halbawi et al. analysed the convergence of the expected delay time as the number of workers grows [5]. Specifically, assuming that a worker's response time follows a Pareto distribution with fundamental delay t_0, Halbawi et al. showed that the expected delay converges to κt_0, where κ is a constant greater than one and is dependent on the workload ratio (number of partitions assigned to a worker).

In this work, we look at the average-case scenario where the taskmaster is able to perform its computation using a smaller set of replies *most of the time*. Specifically, we provide a simple modification to existing optimal schemes and demonstrate that with this modification, the expected delay time converges to t_0, independent of the workload ratio! In other words, we not only alleviate, but also *remove* the effect of stragglers. Additionally, for specific parameters, we also look at improving the convergence rate so that the expected delay time comes close to t_0 with a smaller number of workers.

Before we state our results formally, we provide a short overview of gradient coding.

II. Overview of Gradient Coding

For completeness, we review the gradient coding framework proposed by Tandon et al. [2]. Let D be a dataset that comprises d datapoints $\boldsymbol{z}_1, \cdots, \boldsymbol{z}_d$. Consider an additive function f. Then the required computation for the taskmaster is

$$f(D) \triangleq \sum_{j=1}^{d} f(\mathbf{z}_j). \qquad (1)$$

In machine learning, suppose that we want to learn the parameter $\boldsymbol{\beta}$ of a certain model and the cost function with respect to the dataset D is given by $C(\boldsymbol{\beta}, D)$. To obtain the parameter that minimises the cost, we set the additive function $f(D)$ to be $\nabla C(\boldsymbol{\beta}, D)$.

Since computing $f(D)$ is impractical, we partition D into k parts, $D_1, \cdots, D_k$. If we set $\boldsymbol{g}_j = f(D_j) \triangleq \sum_{z \in D_j} f(\boldsymbol{z})$ for $i \in \{1, \ldots, k\}$ and $\boldsymbol{g} = [\boldsymbol{g}_1, \boldsymbol{g}_2, \cdots, \boldsymbol{g}_k]^T$, then the desired computation can be obtained via

$$f(D) = \mathbf{1}\boldsymbol{g},$$

where $\mathbf{1}$ is the all-ones row vector.

Now, given n workers $\mathsf{W}_1, \cdots, \mathsf{W}_n$, we assign each worker at most w data partitions. To mitigate the effect of s stragglers, it is then necessary to assign the same data partition to more than one worker. Conversely, we look at when we are able to compute $f(D)$ given a set of $n-s$ responses.

Suppose that the worker W_i is assigned to the data partitions $\{D_{i_1}, \cdots D_{i_w}\}$. This means that it is tasked to compute the values $\mathbf{g}_{i_1}, \cdots, \mathbf{g}_{i_w}$. When it completes its computations, it sends the *linear combination* $\mathcal{R}_i = \sum_{j=1}^{w} \lambda_j \mathbf{g}_{i_j}$ to the taskmaster where λ_j's are coefficients predetermined by the taskmaster. We rewrite $\mathcal{R}_i = \sum_{j=1}^{k} \lambda_{i,j} \mathbf{g}_j$ where $\lambda_{i,j} = \lambda_{i_\ell}$ if $j \in \{i_1, \cdots, i_w\}$ and it is 0 otherwise.

Since we want to recover $f(D) = \mathbf{1}\boldsymbol{g}$, the required computation can be retrieved from the replies $\mathcal{R}_{i_1}, \cdots, \mathcal{R}_{i_w}$ if and only if $\mathbf{1}$ belongs to the row space spanned by $\left\{(\lambda_{i_j,1}, \cdots \lambda_{i_j,k}) : j \in \{1, \ldots, w\}\right\}$. We use an $n \times k$ matrix $\boldsymbol{E}$ to record the coefficients $\lambda_{i,j}$'s and summarise our discussion with the following definition.

Definition 1 (Tandon et al. [2])**.** An $(n, s+1, k, w)$-gradient coding matrix (GCM) $\boldsymbol{E}$ is an $n \times k$ complex-valued matrix such that the support of any row of $\boldsymbol{E}$ has size at most w and the all ones vector $\mathbf{1}$ belongs to the row space spanned by any $n-s$ rows of $\boldsymbol{E}$.

Example 1. Let $n = 6, s = 3, k = 3, w = 2$, and consider

$$\boldsymbol{E} = \begin{pmatrix} -2\omega_6 + 1 & 2\omega_6 + 2 & 0 \\ 2\omega_6 + 2 & 0 & -2\omega_6 + 1 \\ 0 & -2\omega_6 + 1 & 2\omega_6 + 2 \\ -2\omega_6 + 4 & 2\omega_6 - 1 & 0 \\ 2\omega_6 - 1 & 0 & -2\omega_6 + 4 \\ 0 & -2\omega_6 + 4 & 2\omega_6 - 1 \end{pmatrix},$$

where ω_6 is a primitive sixth root of unity. To verify that $\boldsymbol{E}$ is an $(n, s+1, k, w)$-GCM, we check that $\mathbf{1}$ is the span of any three rows of $\boldsymbol{E}$.

For example, we check that $\mathbf{1} = \frac{1}{3}(\boldsymbol{r}_1 + \boldsymbol{r}_2 + \boldsymbol{r}_3)$, where $\boldsymbol{r}_i$ is the ith row of $\boldsymbol{E}$. In other words, if $\mathcal{R}_1$, $\mathcal{R}_2$, and $\mathcal{R}_3$ are the responses provided by the first three workers, the taskmaster than immediately obtain $\boldsymbol{g}$ by computing $\frac{1}{3}(\mathcal{R}_1 + \mathcal{R}_2 + \mathcal{R}_3)$. One can verify that this is true for all combinations of three rows.

The following theorem provides a necessary condition for the existence of an $(n, s+1, k, w)$-GCM.

Theorem 1 ([2])**.** *Let n, s, k and w be positive integers. If an $(n, s+1, k, w)$-GCM exists, then $n/(s+1) \geq k/w$. An $(n, s+1, k, w)$-GCM is said to be* optimal *if $n/(s+1) = k/w$.*

A. Previous Results

Tandon et al. [2] introduced the notion of gradient coding and constructed optimal $(n, s+1, k, w)$-GCMs when $n = k$. When $(s+1)$ divides n, an explicit construction was given. Otherwise, the authors provided a randomised construction.

This result was then improved by Raviv et al., who provided explicit constructions using cyclic MDS codes for all s and n, with $k = n$ [4]. Halbawi et al. independently provided explicit constructions of optimal GCMs [5]. Their scheme makes use of row-balanced matrix as a mask and choosing codewords from a suitable Reed-Solomon code for the rows of the encoding matrix. The $(6, 4, 3, 2)$-GCM $\boldsymbol{E}$ in Example 1 is obtained via the construction of Halbawi et al.

Finally, using the GCMs from their construction, Halbawi et al. also derived the expected *delay*, the time for the taskmaster to receive sufficient responses. In the analysis, the processing time of each worker is assumed to be independently and identically distributed under the Pareto distribution with parameters t_0 and ξ. We refer to the quantity t_0 as the *fundamental delay*, that is, the minimum time required for a worker to complete its task.

Proposition 1 ([5])**.** *Let T_i be the time for worker W_i to finish its task. Suppose $T_i \sim \mathrm{Pareto}(t_0, \xi)$ for all i.*

Fix the ratio k/w and set T^H_{delay} to be the time for the taskmaster receive sufficient responses using the GCMs constructed by Halbawi et al.. Then $\lim_{n\to\infty} \mathbb{E}[T^H_{\mathrm{delay}}] = t_0 (w/k)^{-1/\xi}$.

B. Our Contributions

In this work, we focus on reducing the expected delay, assuming the same probability distribution. For fixed values of w, k, n, we set $s = \lfloor wn/k \rfloor - 1$ and Theorem 1 states that the number of replies required is at least $(n-s)$ in the *worst case*. To reduce the expected delay, we modify existing schemes so that less than $(n-s)$ replies are sufficient in most cases.

Our contributions are as follow.

(I) Using a seed GCM in previous works [4], [5], we construct explicit optimal GCMs. For this family of GCMs, we then show that the expected delay approaches the fundamental delay t_0. Our results imply that given a sufficient number of machines, it is possible to distribute the computations such that the effect of stragglers becomes negligible.

(II) For odd n, we provide explicit constructions of $(n, 2, n, 2)$-GCMs such that in most cases $(n+1)/2$ responses are sufficient to recover the gradient. Using these GCMs as seed matrices in our scheme, we demonstrate via simulations that the average delay converges to the fundamental delay more quickly.

III. Repetition Scheme

In this section, we provide a simple construction of GCMs that uses smaller GCMs as building blocks and show that the expected delay is reduced using this construction. First, we introduce the notion of *decodable* and *minimal decodable sets*. Let $\boldsymbol{E}$ be an (n, s, k, w)-GCM. Then the *family of decodable sets* is defined to be

$$\mathcal{S} = \{S \subseteq \{1, 2, \ldots, n\} : \mathbf{1} \text{ is in the span of the rows of } \boldsymbol{E} \text{ corresponding to } S\}. \quad (2)$$

Then the *family of minimal decodable sets* is given by $\mathcal{D} = \{S \in \mathcal{S} : T \notin \mathcal{S} \text{ for all proper subsets } T \text{ of } S\}$.

Example 2. Consider $\boldsymbol{E}$ given in Example 1. Then it can be verified that $\mathcal{D}$ is given by all 3-subsets of $\{1, 2, \ldots, 6\}$.

Example 3. Let $n = 6, s = 4, k = 3, w = 2$ and consider

$$\boldsymbol{E}' = \begin{pmatrix} -\omega_3 + 1 & \omega_3 + 2 & 0 \\ \omega_3 + 2 & 0 & -\omega_3 + 1 \\ 0 & -\omega_3 + 1 & \omega_3 + 2 \\ -\omega_3 + 1 & \omega_3 + 2 & 0 \\ \omega_3 + 2 & 0 & -\omega_3 + 1 \\ 0 & -\omega_3 + 1 & \omega_3 + 2 \end{pmatrix},$$

where ω_3 is a primitive third root of unity. Then it can be verified that $\mathcal{D} = \{\{1,2\},\{1,3\},\{2,3\},\{4,5\},\{4,6\},\{5,6\}\}$. Since any 3-subset of $\{1,2,\ldots,n\}$ necessarily contains a decodable set in $\mathcal{D}$, we verify that $\boldsymbol{E}'$ is also an optimal $(6,4,3,2)$-GCM.

Next, notice that the decodable sets in $\mathcal{D}$ have size two while the decodable sets in Example 2 have size three. Hence, it is likely that the taskmaster is able to commence computation earlier. We formalise this idea and provide a detailed analysis in Section III-A.

Finally, observe that the first three rows form an optimal $(3,2,3,2)$-GCM and is in fact obtained from Halbawi et al.'s construction. The next three rows are simply replicates of the first three rows and this construction can be generalized easily.

Theorem 2 (Repetition Scheme)**.** *Let $b < a$ be two positive integers such that a and b are coprime and $\boldsymbol{E} \in \mathbb{C}^{a\times a}$ be an optimal (a,b,a,b)-GCM. Choose integers n_1, k_1 and set $n = n_1 a$, $s + 1 = n_1 b$, $k = k_1 a$, and $w = k_1 b$.*

Define the block matrix $\boldsymbol{B} \in \mathbb{C}^{n_1 a\times k_1 a}$ as follows:

$$\boldsymbol{B} = n_1 \left\{ \begin{pmatrix} \overbrace{\boldsymbol{E} \;\; \boldsymbol{E} \;\; \cdots \;\; \boldsymbol{E}}^{k_1} \\ \boldsymbol{E} \;\; \boldsymbol{E} \;\; \cdots \;\; \boldsymbol{E} \\ \vdots \;\;\; \vdots \;\;\; \ddots \;\;\; \vdots \\ \boldsymbol{E} \;\; \boldsymbol{E} \;\; \cdots \;\; \boldsymbol{E} \end{pmatrix} \right. \tag{3}$$

Then $\boldsymbol{B}$ is an optimal $(n, s+1, k, w)$-GCM. Furthermore, the minimal decodable sets have size at most $(a - b + 1)$.

Proof. First, consider the following block matrix:

$$\boldsymbol{E}_{k_1} = (\boldsymbol{E}\;\boldsymbol{E}\;\cdots\;\boldsymbol{E}).$$

Since $\boldsymbol{E}$ is an optimal (a,b,a,b)-GCM, the span of any $a-b+1$ rows of $\boldsymbol{E}$ contains the all-ones vector. Since $\boldsymbol{E}_{k_1}$ is consists of k_1 replicates of the matrix $\boldsymbol{E}$ column-wise, then the span of any $a - b + 1$ rows of $\boldsymbol{E}'$ also contains the all-ones vector.

To show that $\boldsymbol{B}$ is an $(n, s+1, k, w)$-GCM, it suffices to show that the span of any $(n - s)$ rows of $\boldsymbol{B}$ contains $\mathbf{1}$. Observe that $\boldsymbol{B}$ consists of n_1 replicates of $\boldsymbol{E}_{k_1}$ row-wise. Since $n - s = n_1(a - b) + 1$, then by pigeonhole principle, any set of $(n - s)$ rows of $\boldsymbol{B}$ contains a set of $(a - b + 1)$ rows of $\boldsymbol{E}_{k_1}$. Since $\mathbf{1}$ is in the span of these $(a-b+1)$ rows, $\mathbf{1}$ is also contained in the span of the $(n - s)$ rows. □

Suppose that an $(n, s+1, k, w)$-GCM $\boldsymbol{B}$ is constructed from an (a,b,a,b)-GCM $\boldsymbol{E}$ using Theorem 2. We call $\boldsymbol{E}$ the *seed matrix* of $\boldsymbol{B}$. Observe that the size of minimal decodable sets for $\boldsymbol{B}$ has the same size as the minimal decodable sets for the smaller seed matrix $\boldsymbol{E}$. In Section IV, we reduce the size of the decodable sets further for the seed matrices.

In the next subsection, we analyse the expected delay for the repetition scheme and demonstrate that this expected delay converges to the fundamental delay.

A. Asymptotic Analysis of Expected Delay

This subsection is devoted to the analysis of the expected waiting time for this scheme. As observed in practice [6], the processing time for each workers is independently and identically distributed by the Pareto distribution with parameters t_0, ξ for some $t_0, \xi > 0$. Formally, when the random variable $X \sim \mathrm{Pareto}(t_0, \xi)$, the cumulative density function of X is given by

$$F_X(t) = 1 - \left(\frac{t_0}{t}\right)^{\xi}, \tag{4}$$

while the probability density function of X is given by

$$f_X(t) = \xi \frac{t_0^{\xi}}{t^{\xi+1}}. \tag{5}$$

Here t_0 represents the *fundamental* or minimum delay of a worker and in practice, we have $1 \le \xi \le 2$ [6]. The following proposition provides the asymptotic approximation of the expected delay resulting from the scheme in Theorem 2.

Proposition 2. *Fix coprime integers a and b and let $\boldsymbol{E}$ be an optimal (a,b,a,b)-GCM. For integer n_1, set $k_1 = 1$ and set $\boldsymbol{B}$ to be the $(n, s+1, k, w)$-GCM constructed from Theorem 2.*

Let T^R_{delay} be the random variable measuring the delay or the time taken by the taskmaster to obtain a decodable set. Then, as $n_1 \to \infty$, the expected delay time converges to t_0. That is,

$$\lim_{n\to\infty} \mathbb{E}(T^R_{\text{delay}}) = t_0.$$

Proof. We reindex the rows of $\boldsymbol{B}$ with $(1,1),(1,2),\ldots,(1,a)$, $(2,1),(2,2),\ldots,(2,a)$, $\ldots$, $(n_1,1),(n_1,2),\ldots,(n_1,a)$. For $i \in \{1,2,\ldots,n_1\}$ and $j \in \{1,2,\ldots,a\}$, we let $T_{(i,j)}$ denote the random variable measuring the processing time of the worker $\mathsf{W}_{(i,j)}$. Then $T_{(i,j)} \sim \mathrm{Pareto}(t_0,\xi)$ and we denote the probability and cumulative density functions of the Pareto distribution with $f_P(t)$ and $F_P(t)$, respectively.

Fix $j \in \{1,2,\ldots,a\}$. Observe that the rows indexed by $(1,j),(2,j),\ldots,(n_1,j)$ are identical. Hence, we are interested in the time required to obtain the response from one of these workers. Set $T_j = \min_{1\le i\le j} T_{(i,j)}$, or T_j is the first order statistic of the set of random variables $\{T_{(1,j)}, T_{(2,j)}, \ldots, T_{(n_1,j)}\}$. Then the PDF of T_j can be computed as

$$\begin{aligned} f_{T_j}(t) &= \frac{n_1!}{(1-1)!(n_1-1)!} f_P(t)(1 - F_P(t))^{n_1-1} \\ &= n_1 \left(\xi \frac{t_0^{\xi}}{t^{\xi+1}}\right) \left(1 - \left(1 - \left(\frac{t_0}{t}\right)^{\xi}\right)\right)^{n_1-1} \\ &= n_1 \left(\xi \frac{t_0^{\xi}}{t^{\xi+1}}\right) \left(\frac{t_0}{t}\right)^{\xi(n_1-1)} = (n_1\xi) \frac{t_0^{n_1\xi}}{t^{n_1\xi+1}}. \end{aligned}$$

In other words, $T_j \sim \text{Pareto}(t_0, n_1\xi)$.

Now, in order to compute the gradient vector, we need at least $(a-b+1)$ rows of $\boldsymbol{E}$. In other words, T_{delay} is the $(a-b+1)$-th order statistics of the set of random variables $\{T_1, T_2, \ldots, T_a\}$. The expected value for this distribution can then be calculated using the formula in [7]. Specifically,

$$\mathbb{E}(T_{\text{delay}}) = t_0 \frac{\Gamma\left(b - \frac{1}{n_1\xi}\right)\Gamma\left(a+1\right)}{\Gamma(b)\Gamma\left(a+1-\frac{1}{n_1\xi}\right)} \tag{6}$$

where $\Gamma(x) = \int_0^\infty t^{x-1}e^{-t}dt$.

Note that

$$\lim_{n_1\to\infty} b - \frac{1}{n_1\xi} = b \text{ and } \lim_{n_1\to\infty} a+1-\frac{1}{n_1\xi} = a+1.$$

Therefore, by the continuity of Gamma function [8], we have

$$\lim_{\epsilon\to\infty} t_0 \frac{\Gamma\left(b - \frac{1}{n_1\xi}\right)\Gamma\left(a+1\right)}{\Gamma(b)\Gamma\left(a+1-\frac{1}{n_1\xi}\right)} = t_0\frac{\Gamma(b)\Gamma(a+1)}{\Gamma(b)\Gamma(a+1)} = t_0. \quad \square$$

Remark 1. Fix the ratio k/w. We consider the family of optimal GCMs constructed by Halbawi et al. [5]. As in Proposition 1, we consider the random variable T_{delay}^H and following our derivation, we have

$$\mathbb{E}(T_{\text{delay}}^H) = t_0 \frac{\Gamma\left(s+1-\frac{1}{\xi}\right)\Gamma(n+1)}{\Gamma(s+1)\Gamma\left(n+1-\frac{1}{\xi}\right)}. \tag{7}$$

This value does not approach t_0 as the difference between the Gamma functions on the numerator and denominator does not approach zero. Instead, Halbawi et al. demonstrated that $\mathbb{E}(T_{\text{delay}}^H)$ converges to $t_0\left(\frac{w}{k}\right)^{-\frac{1}{\xi}}$.

Therefore, the repetition scheme reduces the expected delay by a factor of $\left(\frac{w}{k}\right)^{-\frac{1}{\xi}}$ asymptotically.

B. Numerical Experiments

We perform numerical experiments to corroborate Propositions 1 and 2. Specifically, we set $a = 3$ and $b = 2$ and model the processing time of each worker with the distribution $\text{Pareto}(0.001, 1.1)$ following [5].

For different numbers of workers, we conducted a simulation and obtained the average delay (over 1000 trials) resulting from two different schemes. Specifically, for $n_1 \in \{1, 2, \ldots, 16\}$,, we set $n = k = 3n_1$ and $s + 1 = w = 2n_1$. The first scheme then uses $(n, s+1, k, w)$-GCM constructed by Halbawi et al., while the second scheme uses the $(n, s+1, k, w)$-GCM constructed resulting from Theorem 2 with $\boldsymbol{E}'$ in Example 3 as the seed matrix. We compare the average delay in Figure 1.

Observe that as the number of workers increases, the average delay of the first scheme converges to $(3/2)^{-1/1.1}t_0 \approx 1.45t_0$, corroborating Proposition 1. In contrast, the average delay converges to t_0 verifying Propositon 2.

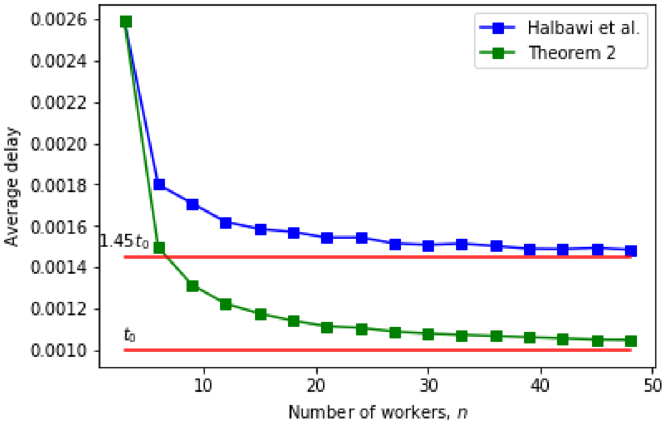

Fig. 1. Comparison of average delay using Halbawi et al.'s scheme and the repetition scheme

IV. Seed Matrices with Smaller Decodable Sets

In the previous section, we used an (a, b, a, b)-GCM as a seed matrix for the repetition scheme and showed that the expected delay converges to the fundamental delay when the number of workers is sufficiently large. In this section, we further reduce the size of minimal decodable sets for the seed matrix so that the expected delay converges as a faster rate.

Now, in previous constructions of optimal GCMs, the authors focussed on ensuring that all minimal decoding sets have size at most $a - b + 1$. Often, these GCMs have the property that all minimal decoding sets have size equal to $a-b+1$. For example, when $a = 5$ and $b = 2$, the construction of Raviv et al. [4] yields the following GCM

$$\boldsymbol{E}_R = \begin{pmatrix} \alpha & 0 & \beta & 0 & 0 \\ \beta & 0 & 0 & \alpha & 0 \\ 0 & \alpha & 0 & \beta & 0 \\ 0 & \beta & 0 & 0 & \alpha \\ 0 & 0 & \alpha & 0 & \beta \end{pmatrix},$$

where $\alpha = \omega_5^4 - \omega_5^2 - 2\omega_5 + 2$, $\beta = -2\omega_5^4 - \omega_5^3 + \omega_5 + 2$, and ω_5 is the primitive fifth root of unity. Then the family of minimal decodable sets for $\boldsymbol{E}_R$ is given by

$$\{\{2,3,4,5\}, \{1,3,4,5\}, \{1,2,4,5\}, \{1,2,3,5\}, \{1,2,3,4\}\}.$$

On the other hand, consider the following matrix

$$\boldsymbol{E}_* = \begin{pmatrix} 2 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & -1 \\ 0 & 0 & 1 & 0 & 2 \end{pmatrix},$$

Then the family of minimal decoding sets is given by

$$\mathcal{D}_* = \{\{1,3,5\}, \{2,4,5\}, \{1,2,3,4\}\}.$$

Observe that each 4-subset of $\{1,2,3,4,5\}$ contains at least one decodable set in $\mathcal{D}_*$. Therefore, $\boldsymbol{E}_*$ is also an optimal $(5,2,5,2)$-GCM.

The above construction can be generalized to all odd a and $b = 2$. Instead of having a minimum decodable sets of size $(a-1)$, the next construction provides an optimal (a, b, a, b)-GCM *three* minimum decodable sets, two of size $(a+1)/2$ and one of size $a-1$.

Theorem 3 (Seed Matrix with $b = 2$)**.** *Let a be an odd integer. Consider the following set of indices*

$$S = \{(2j-1, j), (2j, j) : 1 \le j \le a-1\} \setminus \{(1,1)\},$$

where the arithmetic of indices is taken over modulo a. Construct a matrix $\boldsymbol{E} \in \mathbb{R}^{a \times a}$ as follows

$$\boldsymbol{E}_{i,j} = \begin{cases} 1, & \text{if } (i,j) \in S \\ 2, & \text{if } (i,j) \in \{(1,1),(a,a)\} \\ -1, & \text{if } (i,j) = (a-1,a) \\ 0, & \text{otherwise.} \end{cases}$$

Then the minimum decodable sets are given by

$$\{\{1,3,5,\dots,a-2,a\},\{2,4,6,\dots,a-1,a\},\{1,2,\dots,a-1\}\}.$$

Proof. For each decodable set, we demonstrate that $\mathbf{1}$ is in the span of the corresponding rows. Set $\boldsymbol{r}_i$ to be the ith row of $\boldsymbol{E}$. Then

$$\begin{aligned} \mathbf{1} &= \frac{1}{2}\boldsymbol{r}_1 + \boldsymbol{r}_3 + \cdots + \boldsymbol{r}_{a-2} + \frac{1}{2}\boldsymbol{r}_a \\ &= \boldsymbol{r}_2 + \boldsymbol{r}_4 \cdots + \boldsymbol{r}_{a-1} + \boldsymbol{r}_a \\ &= \boldsymbol{r}_1 - \boldsymbol{r}_2 + 2\boldsymbol{r}_3 - \boldsymbol{r}_4 \cdots + 2\boldsymbol{r}_{a-2} - \boldsymbol{r}_{a-1}. \end{aligned}$$ □

To conclude this section, we perform numerical experiments to examine the effect of reducing the size of the decodable sets. Specifically, we set $a = 5$ and consider the seed matrices $\boldsymbol{E}_R$ and $\boldsymbol{E}_*$ in this section. We then vary $n_1 \in \{1, 2, \dots, 10\}$ with $k_1 = 1$ and construct $(n, s+1, k, w)$-GCMs using Theorem 2 with the two different seed matrices. As before, we model the processing time of each worker with the distribution $\mathrm{Pareto}(0.001, 1.1)$ and compute the average delay from 1000 trials. We then repeated the numerical experiment with $a = 21$. Figure 2 compares the average delays amongst the different schemes.

As observed in Figure 2, even though the schemes have the same asymptotic average delay, the average delay converges to the fundamental delay at a faster rate.

V. Conclusion

In this work, we looked at the problem of stragglers in the distributed computation of the gradient function in machine learning. By modifying existing schemes, we are able to compute the gradient with significantly less replies in most cases and hence, reduce the expected delay.

Of significance, we used seed GCMs in previous works [4], [5] to construct explicit optimal GCMs. For this family of GCMs, we then showed that the expected delay approaches the fundamental delay t_0 (see Theorem 2 and Proposition 2). Our results therefore imply that given a sufficient number of machines, it is possible to distribute the computations such that the effect of stragglers becomes negligible.

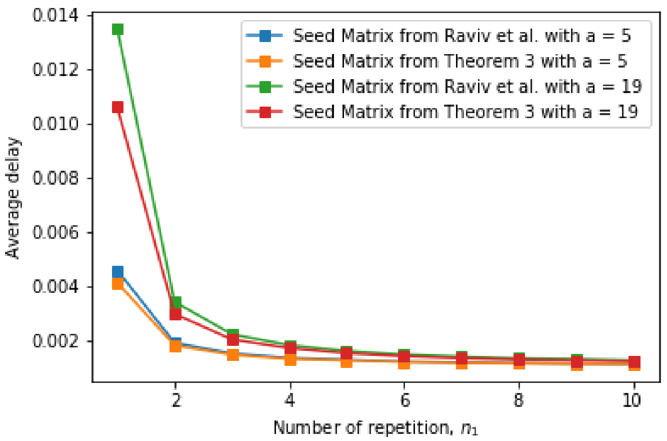

Fig. 2. Comparison of average delay using various seed matrices

In Section IV, we then provided an explicit construction of $(n, 2, n, 2)$-GCMs such that in most cases $(n+1)/2$ responses are sufficient to recover the gradient. As illustrated by Figure 2, suitable modifications of the seed matrices result in a significant improvement in delay. Therefore, it is of practical interest to design seed matrices with smaller decodable sets for $b > 2$.

References

[1] A. Harlap, H. Cui, W. Dai, J. Wei, G. R. Ganger, P. B. Gibbons, G. A. Gibson, and E. P. Xing, "Addressing the straggler problem for iterative convergent parallel ml," in *Proceedings of the Seventh ACM Symposium on Cloud Computing*, ser. SoCC '16. New York, NY, USA: ACM, 2016, pp. 98–111. [Online]. Available: http://doi.acm.org/10.1145/2987550.2987554

[2] 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. International Convention Centre, Sydney, Australia: PMLR, 06–11 Aug 2017, pp. 3368–3376. [Online]. Available: http://proceedings.mlr.press/v70/tandon17a.html

[3] J. Cipar, Q. Ho, J. K. Kim, S. Lee, G. R. Ganger, G. Gibson, K. Keeton, and E. Xing, "Solving the straggler problem with bounded staleness," in *Proceedings of the 14th USENIX Conference on Hot Topics in Operating Systems*, ser. HotOS'13. Berkeley, CA, USA: USENIX Association, 2013, pp. 22–22. [Online]. Available: http://dl.acm.org/citation.cfm?id=2490483.2490505

[4] N. Raviv, I. Tamo, R. Tandon, and A. G. Dimakis, "Gradient coding from cyclic MDS codes and expander graphs," 2017. [Online]. Available: arXiv:abs/1707.03858

[5] W. Halbawi, N. A. Ruhi, F. Salehi, and B. Hassibi, "Improving distributed gradient descent using reed-solomon codes," 2017. [Online]. Available: arXiv:abs/1706.05436

[6] M. Harchol-Balter, "The effect of heavy-tailed job size distributions on computer system design." in *Proc. of ASA-IMS Conf. on Applications of Heavy Tailed Distributions in Economics, Engineering and Statistics*, 1999.

[7] K. Vännman, "Estimators based on order statistics from a pareto distribution," *Journal of the American Statistical Association*, vol. 71, no. 355, pp. 704–708, 1976. [Online]. Available: https://doi.org/10.1080/01621459.1976.10481551

[8] L. M. Hall, "Special function," 1995. [Online]. Available: http://web.mst.edu/ lmhall/SPFNS/sfch2.pdf