Timely Updates in Distributed Computation Systems with Stragglers

Baturalp Buyukates Sennur Ulukus
Department of Electrical and Computer Engineering
University of Maryland, College Park, MD 20742
baturalp@umd.edu *ulukus@umd.edu*

***Abstract*—We consider a status update system in which the update packets need to be processed to extract the embedded useful information. The source node sends the acquired information to a computation unit (CU) which consists of a master node and n worker nodes. The master node distributes the received computation task to the worker nodes. Upon computation, the master node aggregates the results and sends them back to the source node to keep it *updated*. We study the age performance of uncoded and coded (repetition coded, MDS coded, and multi-message MDS (MM-MDS) coded) schemes in the presence of stragglers under i.i.d. exponential transmission delays and i.i.d. shifted exponential computation times. We show that asymptotically MM-MDS coded scheme outperforms the other schemes. Finally, we characterize the age-optimal codes.**

I. Introduction

Age of information metric has been widely studied as a timeliness metric in real-time systems producing time-sensitive information. Most of the existing literature on age of information assumes small sized status update packets and studies the queueing-theoretic framework under various arrival/service profiles and optimization, scheduling and energy harvesting settings (see the survey in [1]).

In contrast, in this paper, we consider status update systems that are prevalent in emerging data intensive applications such as autonomous vehicle systems which involve more complex update settings that require further processing to extract the embedded information; e.g., vehicles take pictures/videos of the scene and send them to a server system which processes them to generate a simple eventual update such as "reduce speed". References [2]–[8] consider such computation-intensive status update packets. Common to all these works is the fact that they consider a single computation server per job.

In this work, we consider a multi-server system with distributed computation capability to process the computation-intensive update packets. A source node uploads status update packets to a computation unit (CU) which consists of a single master node and n worker nodes (see Fig. 1). We assume that the required computation is a linear operation such as large matrix multiplication. This brings up computation distribution and scheduling among the worker nodes which has been extensively studied in the literature [9]–[21].

This work was supported by NSF Grants CCF 17-13977 and ECCS 18-07348.

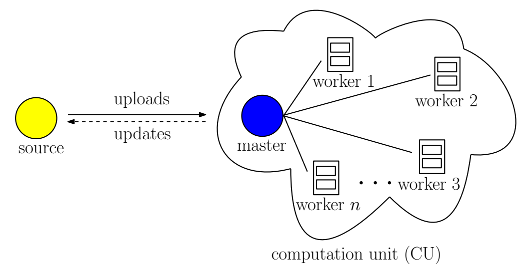


Fig. 1. System model with a single source node and a computation unit (CU) that consists of a master node and n identical worker nodes.

In our model, the master node distributes the arriving computation task to n worker nodes using uncoded or coded schemes. Once the master node collects sufficiently many results from the worker nodes to decode the computation result, it *updates* the source node. Unlike the existing distributed computation literature which uses metrics such as expected overall runtime, straggler thresholds, and so on, to evaluate the performance of distributed computation systems, our goal is to investigate the timeliness of these distributed computation systems based on the age of information metric. We study well-known uncoded and coded computation distribution algorithms to design a system which can tolerate and combat stragglers, as well as, achieve a minimum age of information.

We derive the average age for uncoded and coded schemes and show that asymptotically multi-message MDS (MM-MDS) coded scheme outperforms the uncoded, repetition coded and MDS coded schemes. Our results indicate that given that the source node and the CU implement zero-wait and dropping policies, respectively, when the transmission delays are i.i.d. exponentials and computation times are i.i.d. shifted exponentials, for large n, minimizing age of information is equivalent to minimizing the computation time. Finally, we find the age-optimal codes that minimize the average age.

II. System Model and Age Metric

The source node adopts a zero-wait policy in which it sends the next update when the current one reaches the CU. Random variable D denotes the i.i.d. transmission delays experienced by the update packets from the source node to the CU and is exponentially distributed with rate λ. That is, computation tasks arrive at the CU following a Poisson process with rate λ. Here, we use status update packet and computation task interchangeably. The CU implements a dropping policy such that arriving update packets can only enter the CU if the CU is idle at the time of their arrival.

The master node distributes the computation tasks that successfully enter the CU to n worker nodes by adopting uncoded or coded distribution algorithms. We analyze the effects of these schemes on the timeliness of the computations.[1]

Each worker node performs the computation and sends the result back to the master node. Computation times of the workers are i.i.d. and we assume a mother runtime distribution as in [9]. This distribution corresponds to the computation time, including the time spent in communicating the inputs and the outputs of the computation with the master node, when the whole computation is performed by a single worker, X, and has a shifted exponential distribution with (c, μ) where $c > 0$. The constant shift makes sure that computation times cannot go below a certain value and the exponential part constitutes the tail of the distribution. When the update packet is divided into m subpackets, the computation time of each subpacket has the sped-up version of the overall distribution, i.e., shifted exponential with $\left(\frac{c}{m}, m\mu\right)$ [9].

When the master node receives sufficiently many responses from the worker nodes, it aggregates the results, *updates* the source node and waits for the next packet arrival. This idle waiting time is denoted by Z. We neglect the transmission delay from the CU back to the source node after computation as the size of the initial packet is in general much larger than the resulting simple update packet after computation.

To quantify the timeliness, we use the age of information metric. At time t, age at the source node, is a random process $\Delta(t) = t - u(t)$ where $u(t)$ is the timestamp of the most recent update received by the source node. The metric we use, time averaged age, is $\Delta = \lim_{\tau\to\infty} \frac{1}{\tau}\int_0^\tau \Delta(t)dt$.

III. Age of Information Analysis

We denote the packets that find the CU idle and thus go into service as the successful packets. Let T_{j-1} and T'_{j-1} denote the time at which the jth successful packet is generated at the source node and is received by the CU, respectively. Thus, $D_j = T'_{j-1} - T_{j-1}$. Let Y denote the update cycle at the CU, i.e., the time in between two consecutive successful arrivals, and $Y_j = T'_j - T'_{j-1}$. Update cycle Y_j consists of computation (service) time S_j and idle waiting time Z_j. We note that Z_j, S_j and D_j are mutually independent, and sequences $\{D_1, D_2, \dots\}$ and $\{S_1, S_2, \dots\}$ form i.i.d. processes.

We observe that Z is stochastically equal to the transmission delay D, i.e., interarrival time at the CU, due to the memoryless property. Computation time S changes depending on the task distribution algorithm adopted by the master node. Let $X_{k:n}$ denote the kth smallest of $X_1, \dots, X_n$. For a shifted exponential random variable X with (c, μ), we have [22]

$$\mathbb{E}[X_{k:n}] = c + \frac{1}{\lambda}(H_n - H_{n-k}), \tag{1}$$

[1]We note that encoding and decoding times are out of the scope of this work as in [9], [15]–[17]. That is, we do not consider encoding and decoding complexities when comparing uncoded and coded schemes.

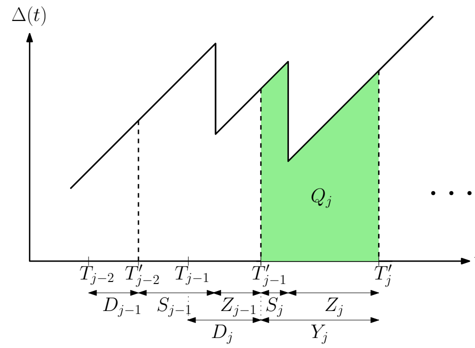


Fig. 2. Sample age evolution $\Delta(t)$ at the source node. Successful updates are indexed by j. The jth successful update leaves the source node at T_{j-1} and arrives at the CU at T'_{j-1}. Update cycle at the CU is the time in between two successive arrivals and is equal to $Y_j = S_j + Z_j = T'_j - T'_{j-1}$.

$$Var[X_{k:n}] = \frac{1}{\lambda^2}(G_n - G_{n-k}), \tag{2}$$

where $H_n = \sum_{j=1}^n \frac{1}{j}$ and $G_n = \sum_{j=1}^n \frac{1}{j^2}$.

The metric we use, long term average age, is the average area under the age curve which is given by [23]

$$\Delta = \limsup_{n\to\infty} \frac{\frac{1}{n}\sum_{j=1}^n Q_j}{\frac{1}{n}\sum_{j=1}^n Y_j} = \frac{\mathbb{E}[Q]}{\mathbb{E}[Y]}, \tag{3}$$

where Q denotes the shaded area and Y is its length in Fig. 2. From Fig. 2, we find $\mathbb{E}[Q] = \mathbb{E}[D](\mathbb{E}[S] + \mathbb{E}[Z]) + \frac{1}{2}\mathbb{E}[(S+Z)^2] + \mathbb{E}[S](\mathbb{E}[S] + \mathbb{E}[Z])$ and $\mathbb{E}[Y] = \mathbb{E}[S] + \mathbb{E}[Z]$. Then,

$$\Delta = \frac{\mathbb{E}[Q]}{\mathbb{E}[Y]} = \mathbb{E}[D] + \mathbb{E}[S] + \frac{\mathbb{E}[Y^2]}{2\mathbb{E}[Y]}. \tag{4}$$

The first term in (4) reflects the fact that arriving packets at the CU have aged on average by $\mathbb{E}[D]$. Our goal is to minimize the average age given in (4) by adjusting computation (service) time S at the CU through different task distribution algorithms.

A. Uncoded Scheme

The master node divides the received status update packet into n subpackets, one for each worker node. From the mother runtime distribution, computation time at each worker $\tilde{X}$ follows a shifted exponential distribution with $\left(\frac{c}{n}, n\mu\right)$ as each worker node performs a part of the overall computation. Thus, the master node needs to collect all of the results so that $S = \tilde{X}_{n:n}$. From (1)-(2) along with (4) we find the average age when the uncoded scheme is utilized, Δ_{unc}, as

$$\begin{aligned}\Delta_{unc} =& \frac{1}{\lambda} + \frac{c}{n} + \frac{H_n}{n\mu} \\ &+ \frac{\left(\frac{c}{n} + \frac{H_n}{n\mu}\right)^2 + \frac{G_n}{n^2\mu^2} + \frac{2}{\lambda}\left(\frac{c}{n} + \frac{H_n}{n\mu}\right) + \frac{2}{\lambda^2}}{2\left(\frac{c}{n} + \frac{H_n}{n\mu} + \frac{1}{\lambda}\right)}. \end{aligned}\tag{5}$$

The following theorem states the asymptotic average age performance of the uncoded scheme as n increases.

Theorem 1 *With i.i.d. exponential transmission delays and i.i.d. shifted exponential computation times, the average age of the uncoded distribution scheme for large n is $\frac{2}{\lambda} + O\left(\frac{\log n}{n}\right)$.*

The proof of Theorem 1 follows from the fact that for large n, we have $H_n \approx \log n$ and $G_n \approx \frac{\pi^2}{6}$. The constant $\frac{2}{\lambda}$ in the result reflects the sum of $\mathbb{E}[D] = \frac{1}{\lambda}$, the expected delay packets experience on the way from the source to the CU, and $\mathbb{E}[Z] = \frac{1}{\lambda}$, the expected waiting time for a new packet at the CU when it is idle. The $O\left(\frac{\log n}{n}\right)$ term in the result reflects that the average age decreases with n. The uncoded scheme is prone to large delays due to straggling nodes as the master node needs all of the computation results to extract the embedded information. To cope with this issue, redundant computation tasks may be created via coding [9]–[21]. In what follows we analyze the effects of repetition coded, MDS coded and MM-MDS coded schemes on the average age.

B. Repetition Coded Scheme

The packet is divided into k equal sized subpackets where $k \leq n$ and each subpacket has $\frac{n}{k}$ replicas. Thus, computation times at worker nodes, $\tilde{X}$, follow a shifted exponential distribution with $\left(\frac{c}{k}, k\mu\right)$. Since there are $\frac{n}{k}$ workers for each of the k subpackets, the computation time of each subpacket is $\bar{X} = \tilde{X}_{1:\frac{n}{k}}$, where $\bar{X}$ follows a shifted exponential distribution with $\left(\frac{c}{k}, n\mu\right)$. The master node needs k distinct results so that $S = \bar{X}_{k:k}$. Using (4) along with the moments in (1)-(2), we find the average age of the repetition coded scheme, Δ_{rep}, as

$$\Delta_{rep} = \frac{1}{\lambda} + \frac{c}{k} + \frac{H_k}{n\mu} + \frac{\left(\frac{c}{k} + \frac{H_k}{n\mu}\right)^2 + \frac{G_k}{n^2\mu^2} + \frac{2}{\lambda}\left(\frac{c}{k} + \frac{H_k}{n\mu}\right) + \frac{2}{\lambda^2}}{2\left(\frac{c}{k} + \frac{H_k}{n\mu} + \frac{1}{\lambda}\right)}. \quad (6)$$

The following theorem states the asymptotic average age performance of the repetition coded scheme as n increases.

Theorem 2 *With i.i.d. exponential transmission delays and i.i.d. shifted exponential computation times, the average age of the $\frac{n}{k}$-repetition coded scheme for large n with $k = \alpha n$ where $0 < \alpha \leq 1$ is $\frac{2}{\lambda} + O\left(\frac{\log n}{n}\right)$.*

The proof of Theorem 2 follows similarly from that of Theorem 1. Here, we observe that although a coding scheme is implemented, asymptotically, we achieve the same average age performance as the uncoded scheme. Thus, repetition coded scheme is asymptotically no better than the uncoded scheme. Next, we analyze the performance of the MDS coded schemes.

C. MDS Coded Scheme

The update packet is divided into k equal sized subpackets where $k < n$. From these k subpackets a total of n subpackets are created by using $n - k$ redundant subpackets. Thus, each worker node completes its computation in $\tilde{X}$ which is a shifted exponential with $(\frac{c}{k}, k\mu)$. Since k computation results are enough for the master to extract the information, $S = \tilde{X}_{k:n}$. Using this along with (1)-(2) in (4), we find the average age when an (n, k)-MDS code is implemented, Δ_{mds}, as

$$\Delta_{mds} = \frac{1}{\lambda} + \frac{c}{k} + \frac{H_n - H_{n-k}}{k\mu} + \frac{\left(\frac{c}{k} + \frac{H_n - H_{n-k}}{k\mu}\right)^2 + \frac{G_n - G_{n-k}}{k^2\mu^2}}{2\left(\frac{c}{k} + \frac{H_n - H_{n-k}}{k\mu} + \frac{1}{\lambda}\right)} + \frac{\frac{2}{\lambda}\left(\frac{c}{k} + \frac{H_n - H_{n-k}}{k\mu}\right) + \frac{2}{\lambda^2}}{2\left(\frac{c}{k} + \frac{H_n - H_{n-k}}{k\mu} + \frac{1}{\lambda}\right)}. \quad (7)$$

The following theorem gives the asymptotic average age performance of the MDS coded scheme for large n.

Theorem 3 *With i.i.d. exponential transmission delays and i.i.d. shifted exponential computation times, the average age of the (n, k)-MDS coded scheme for large n with $k = \alpha n$ where $0 < \alpha < 1$ is $\frac{2}{\lambda} + O\left(\frac{1}{n}\right)$.*

The proof of Theorem 3 follows similarly to that of Theorem 1 and is detailed in [24]. We observe that the average age in Theorem 3 has a $O\left(\frac{1}{n}\right)$ term as opposed to $O\left(\frac{\log n}{n}\right)$ terms in Theorems 1 and 2. Thus, for large n, MDS coded scheme outperforms repetition coded and uncoded schemes in terms of average age performance. Up to now, we have investigated uncoded and coded schemes in which each worker node is assigned one subtask to compute. In the next subsection, we consider the performance of MDS coded scheme when each worker is given multiple subtasks to compute [13], [16], [17].

D. Multi-message MDS (MM-MDS) Coded Scheme

Each worker node is assigned ℓ subpackets to compute in each update cycle and we implement an $(n\ell, k)$-MDS code where $k < n\ell$. Thus, the overall update packet is divided into k subtasks and from these subtasks $n\ell - k$ redundant subtasks are generated such that the master node needs k computation results to extract the embedded information. Unlike regular MDS coded scheme in which $\ell = 1$, in this scheme faster workers can perform multiple computations to aid the overall computation time. Hence, we utilize partial stragglers, also called non-persistent stragglers [16], i.e., worker nodes that finish some portion of the subtasks that are assigned to them.

Computation time of a subtask at each worker, $\tilde{X}$, has a shifted exponential distribution with $\left(\frac{c}{k}, k\mu\right)$. Following the model in [16], we assume that the duration of each subtask computation performed by a worker node during an update cycle is identical. In other words, if a worker finishes m of the ℓ subtasks during an update cycle, duration of each computation is identical. Therefore, the time it takes for a worker node to perform m computations, $m\tilde{X}$, is also a shifted exponential with $(\frac{mc}{k}, \frac{k\mu}{m})$.

In what follows, the mth level refers to the set of subtasks that are performed by the corresponding worker nodes upon completion of their first $m-1$ subtasks. Let k_m denote the number of subtasks computed in the mth level upon completion of the overall task during an update cycle. We then have $\sum_{m=1}^{\ell} k_m = k$. Fig. 3 shows an example for $n = 10$, $k = 7$, and $\ell = 3$. Here, each column represents the computation times of ℓ subtasks that a worker node is assigned and row m represents the computation times of the

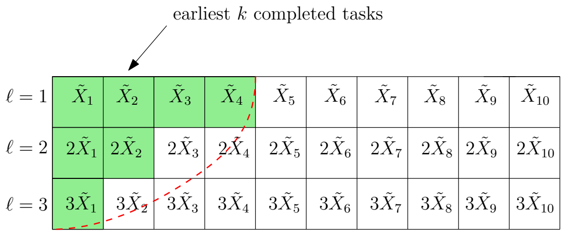


Fig. 3. The earliest k computed tasks for $n=10$, $\ell=3$, and $k=7$.

mth level subtasks. Without loss of generality, we order level one, i.e., $\tilde{X}_1$ in Fig. 3 is the smallest computation time of a level 1 subtask and $\tilde{X}_{10}$ is the largest one. Correspondingly, all other levels are ordered as well. Hence, column i in Fig. 3 in fact shows the computation times of the ith fastest worker node, where $i=1,\dots,n$. Here, we observe that by the time the earliest $k=7$ computations are finished, the fastest worker completed three subtasks, the second fastest worker completed two subtasks, the third and fourth fastest workers completed one subtask each, and the remaining six workers completed zero subtasks. That is, we have $k_1=4$, $k_2=2$ and $k_3=1$.

When we have ℓ levels, with $k_m=\alpha_m n$ with $0<\alpha_m<1$ for large n we find the following relationship between α_ms

$$\frac{1}{(1-\alpha_{m-1})^{m-1}}=e^{\mu c}\frac{1}{(1-\alpha_m)^m}, \tag{8}$$

with $\sum_{m=1}^{\ell}\alpha_m=\ell\alpha$. The proof of this statement, omitted due to space limitations here, is provided in [24]. We note that if after some level $m>m_0$, none of the level m subtasks are finished, then, $\alpha_m=0$ for all $m>m_0$ and (8) holds for all nonzero α_ms. As a direct consequence of (8), we see that the time it takes to receive the earliest k computation results is equivalent to the time it takes to receive k_m from level m such that $k_m=\alpha_m n$ and α_ms satisfy (8) and $\sum_{m=1}^{\ell}\alpha_m=\ell\alpha$. We then have $S=\tilde{X}_{k_1:n}$. Hence, the average age when the MM-MDS coded scheme is implemented with ℓ subpackets at each node, Δ_{mm-mds}, can be computed using (4) as follows

$$\begin{aligned}\Delta_{mm-mds}=&\frac{1}{\lambda}+\frac{c}{k}+\frac{H_n-H_{n-k_1}}{k\mu}\\&+\frac{\left(\frac{c}{k}+\frac{H_n-H_{n-k_1}}{k\mu}\right)^2+\frac{G_n-G_{n-k_1}}{k^2\mu^2}}{2\left(\frac{c}{k}+\frac{H_n-H_{n-k_1}}{k\mu}+\frac{1}{\lambda}\right)}\\&+\frac{\frac{2}{\lambda}\left(\frac{c}{k}+\frac{H_n-H_{n-k_1}}{k\mu}\right)+\frac{2}{\lambda^2}}{2\left(\frac{c}{k}+\frac{H_n-H_{n-k_1}}{k\mu}+\frac{1}{\lambda}\right)}.\end{aligned} \tag{9}$$

where $k_m=\alpha_m n$ and α_ms satisfy (8) and $\sum_{m=1}^{\ell}\alpha_m=\ell\alpha$.

The following theorem gives the asymptotic average age performance of the MM-MDS coded scheme for large n.

Theorem 4 *With i.i.d exponential transmission delays and i.i.d. shifted exponential computation times, the average age of the MM-MDS coded scheme with load ℓ, for large n with $k_m=\alpha_m n$ where $0<\alpha_m<1$, $m=1,\dots,\ell$, is $\frac{2}{\lambda}+O\left(\frac{1}{\ell n}\right)$.*

The proof of Theorem 4 is omitted due to space limitations here and is provided in [24]. We note that compared to the MDS coded scheme where we have $O\left(\frac{1}{n}\right)$, here in the MM-MDS coded scheme, we have $O\left(\frac{1}{\ell n}\right)$. Thus, for large n, the best asymptotic performance is achieved when MM-MDS coded scheme is implemented.

In the next section, we optimize the performance of the discussed coded schemes through the selection of k.

IV. Optimizing Age by Parameter Selection

In this section, we consider the optimization of the parameter k which depends on n linearly as $k=\alpha n\ell$, where $\ell=1$ for repetition and MDS coded schemes, and $\ell>1$ for MM-MDS coded scheme. This optimization is equivalent to the optimization of the parameter α. We first provide the following theorem which shows that, in our model, age minimization translates into computation (service) time minimization which is not in general the case in age optimization problems. The proof of Theorem 5 is omitted due to space limitations here and is provided in [24].

Theorem 5 *When the transmission delays are i.i.d. exponentials and computation times at each worker are i.i.d. shifted exponentials under the dropping policy at the CU, for large n, minimization of the average age is equivalent to minimization of the average computation time.*

For large n, average computation time is given, for the repetition coded scheme, by

$$\mathbb{E}[S_{rep}]=\frac{c}{\alpha n}+\frac{1}{\mu n}\log(\alpha n), \tag{10}$$

for the MDS coded scheme, by

$$\mathbb{E}[S_{mds}]=\frac{c}{\alpha n}+\frac{1}{\mu\alpha n}\log\left(\frac{1}{1-\alpha}\right), \tag{11}$$

and for the MM-MDS coded scheme, by

$$\mathbb{E}[S_{mm-mds}]=\frac{c}{\alpha n\ell}+\frac{1}{\mu\alpha n\ell}\log\left(\frac{1}{1-\alpha_1}\right), \tag{12}$$

where in (12) α_ms satisfy (8) and $\sum_{m=1}^{\ell}\alpha_m=\ell\alpha$.

Reference [9] finds the optimal k, or equivalently the optimal α, for repetition coded and MDS coded schemes when k is linear in n, i.e., $k=\alpha n$. In [9, Lemma 1], (10) is minimized and the optimum α is found as

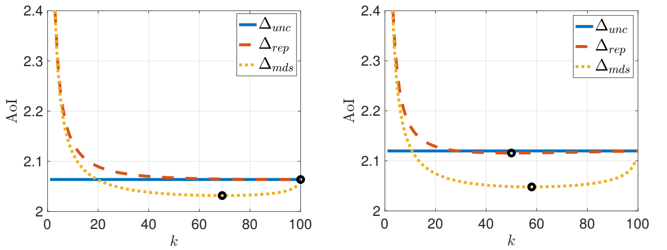
$$\alpha^*=\begin{cases}1, & c\mu\geq 1\\ c\mu, & c\mu<1,\end{cases} \tag{13}$$

for large n which is also average age optimum from Theorem 5 for $\frac{n}{k}$-repetition coded scheme. Note that, for $c\mu\geq 1$, the optimal repetition coded scheme is in fact the uncoded scheme.

For the (n,k)-MDS coded scheme, (11) is minimized in [9, Lemma 2] and it is shown that the optimum α is

$$\alpha^*=1+\frac{1}{W_{-1}(-e^{-\mu c-1})}, \tag{14}$$

for large n, which is also average age optimal from Theorem 5. Here, $W_{-1}(\cdot)$ is the lower branch of Lambert W function [25].



(a) (b)

Fig. 4. Δ_{unc}, Δ_{rep} and Δ_{mds} for varying k with $n = 100$, and $(\lambda, c) = 1$: (a) When $\mu = 1$, (b) when $\mu = 0.5$. Symbol $\circ$ marks the optimal k values.

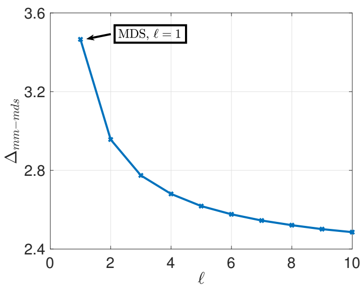


Fig. 5. Δ_{mm-mds}, as a function of load ℓ for $(n\ell, k^*)$-MDS code with $n = 100$, $\mu = 0.01$, and $(\lambda, c) = 1$.

To minimize the age in the MM-MDS coded scheme, we need to solve the following optimization problem:

$$\begin{aligned} \min_{0<\alpha,\alpha_1,\cdots,\alpha_\ell<1} \quad & \frac{c}{\alpha} + \frac{1}{\mu\alpha}\log\left(\frac{1}{1-\alpha_1}\right) \\ \text{s.t.} \quad & \frac{1}{(1-\alpha_{m-1})^{m-1}} = \frac{e^{\mu c}}{(1-\alpha_m)^m}, \ m = 2, \dots, \ell \\ & \sum_{m=1}^{\ell} \alpha_m = \ell\alpha \end{aligned} \tag{15}$$

We note that unlike the repetition coded and MDS coded schemes, in the MM-MDS coded scheme, the problem (15) is more complicated. The optimization in (15) is over α and all α_ms. Here, a closed-form expression for k^*, or equivalently α^*, is not available unlike the former two cases. We solve the problem in (15) in the next section using numerical methods.

V. Numerical Results

In this section, we provide simple numerical results.

First, we show the age performance of the uncoded, repetition coded and MDS coded schemes when $n = 100$, and $(\lambda, c) = 1$ for $\mu = 1$ and $\mu = 0.5$, for varying k in Figs. 4(a) and 4(b), respectively. We observe that in both Figs. 4(a) and 4(b), MDS coded scheme performs the best. Optimal k values for the MDS coded scheme in these cases are $k^* = 69$ and $k^* = 58$, respectively. Moreover, we observe that when $\mu = 1$ optimal k for repetition coded scheme is in fact $k^* = 100$ whereas when $\mu = 0.5$, we get $k^* = 50$ which are in line with (13). We also observe in Fig. 4(b) that repetition coded scheme beats the uncoded scheme when $c\mu < 1$. However, as seen in Fig. 4(a) when $c\mu \geq 1$, repetition coded scheme performs the same as the uncoded scheme.

Next, we consider the MM-MDS coded scheme. Fig. 5 shows the improvement in the average age of MDS coded scheme as a function of load ℓ with $n = 100$, $\mu = 0.01$, and $(\lambda, c) = 1$ when age-optimal k values are used for each ℓ. We note that when $\ell = 1$ we recover the performance of the regular MDS coded scheme and observe that when multiple subpackets are assigned to each worker, we achieve a lower average age than all the other schemes discussed.

References

[1] Y. Sun, I. Kadota, R. Talak, and E. Modiano. Age of information: A new metric for information freshness. *Synthesis Lectures on Communication Networks*, 12(2):1–224, December 2019.

[2] A. Alabbasi and V. Aggarwal. Joint information freshness and completion time optimization for vehicular networks. December 2018. Available on arXiv:1811.12924.

[3] Q. Kuang, J. Gong, X. Chen, and X. Ma. Age-of-information for computation-intensive messages in mobile edge computing. In *IEEE WCSP*, October 2019.

[4] J. Gong, Q. Kuang, X. Chen, and X. Ma. Reducing age-of-information for computation-intensive messages via packet replacement. In *IEEE WCSP*, October 2019.

[5] A. Arafa, R. D. Yates, and H. V. Poor. Timely cloud computing: Preemption and waiting. In *Allerton Conference*, September 2019.

[6] X. Song, X. Qin, Y. Tao, B. Liu, and P. Zhang. Age based task scheduling and computation offloading in mobile-edge computing systems. In *IEEE WCNC*, April 2019.

[7] J. Zhong, W. Zhang, R. D. Yates, A. Garnaev, and Y. Zhang. Age-aware scheduling for asynchronous arriving jobs in edge applications. In *IEEE INFOCOM*, April 2019.

[8] P. Zou, O. Ozel, and S. Subramaniam. Trading off computation with transmission in status update systems. In *IEEE PIMRC*, September 2019.

[9] K. Lee, M. Lam, R. Pedarsani, D. Papailiopoulos, and K. Ramchandran. Speeding up distributed machine learning using codes. *IEEE Transactions on Information Theory*, 64(3):1514–1529, March 2018.

[10] R. Tandon, A. Lei, G. Dimakis, and N. Karampatziakis. Gradient coding: Avoiding stragglers in distributed learning. In *ICML*, August 2017.

[11] A. B. Das, L. Tang, and A. Ramamoorthy. C^3LES: Codes for coded computation that leverage stragglers. In *IEEE ITW*, November 2018.

[12] S. Dutta, M. Fahim, F. Haddadpour, H. Jeong, V. Cadambe, and P. Grover. On the optimal recovery threshold of coded matrix multiplication. *IEEE Transactions on Information Theory*, 66(1):278–301, January 2020.

[13] S. Kiani, N. Ferdinand, and S. C. Draper. Exploitation of stragglers in coded computation. In *IEEE ISIT*, June 2018.

[14] N. Ferdinand and S. C. Draper. Hierarchical coded computation. In *IEEE ISIT*, June 2018.

[15] M. M. Amiri and D. Gunduz. Computation scheduling for distributed machine learning with straggling workers. In *IEEE ICASSP*, May 2019.

[16] E. Ozfatura, D. Gunduz, and S. Ulukus. Speeding up distributed gradient descent by utilizing non-persistent stragglers. In *IEEE ISIT*, July 2019.

[17] E. Ozfatura, D. Gunduz, and S. Ulukus. Gradient coding with clustering and multi-message communication. In *IEEE DSW*, June 2019.

[18] E. Ozfatura, S. Ulukus, and D. Gunduz. Distributed gradient descent with coded partial gradient computations. In *IEEE ICASSP*, May 2019.

[19] M. F. Aktas and E. Soljanin. Straggler mitigation at scale. *IEEE/ACM Transactions on Networking*, 27(6):2266–2279, December 2018.

[20] C. S. Yang, R. Pedarsani, and A. S. Avestimehr. Timely coded computing. In *IEEE ISIT*, July 2019.

[21] K. Duffy and S. Shneer. MDS coding is better than replication for job completion times. July 2019. Available on arXiv: 1907.11052.

[22] J. Zhong, E. Soljanin, and R. D. Yates. Status updates through multicast networks. In *Allerton Conference*, October 2017.

[23] E. Najm, R. D. Yates, and E. Soljanin. Status updates through M/G/1/1 queues with HARQ. In *IEEE ISIT*, June 2017.

[24] B. Buyukates and S. Ulukus. Timely distributed computation with stragglers. *IEEE Transactions on Communications*, 68(9):5273–5282, September 2020.

[25] R. M. Corless, G. H. Gonnet, D. E. G. Hare, D. J. Jeffrey, and D. E. Knuth. On the Lambert W function. *Advances in Computational Mathematics*, 5(1):329–359, December 1996.