Distributed stochastic inertial-accelerated methods with delayed derivatives

Yangyang Xu, Yibo Xu, Yonggui Yan, Jie Chen

partly supported by NSF Award #2053493 and IBM

INFORMS Optimization Society Conference

March 13, 2022

Motivations

- data too big to fit on a single machine (e.g., ImageNet about 150 GB)
- data collected by different agents (e.g., medical data, financial data)
- accessibility to high-performance computing resources
- slow convergence of certain reliable methods

Goals

- design new algorithms for distributed optimization
- fast convergence and high parallelization speed-up

Outline

1. Problem formulation and examples

2. Proposed algorithm and convergence rate results

3. Numerical results for phase retrieval and machine learning

Stochastic problem formulation

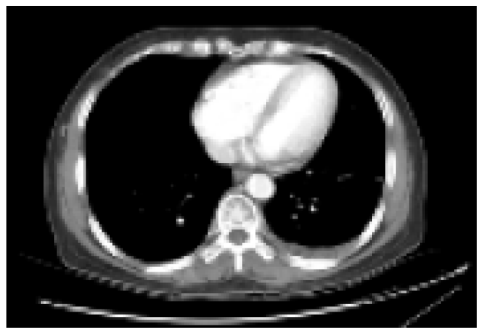
$$\phi^* = \min_{\mathbf{x}\in\mathbb{R}^n} \ \phi(\mathbf{x}) := F(\mathbf{x}) + r(\mathbf{x}), \ \text{with } F(\mathbf{x}) := \mathbb{E}_\xi[f(\mathbf{x};\xi)]$$

- $\mathbf{x}$ decision variable (e.g., parameters of a neural network)
- ξ a random variable that can represent a stochastic scenario or a data point
- reduces to the finite-sum problem when $\xi \sim \{\xi_1, \ldots, \xi_N\}$.
- F is nonconvex and can be smooth or nonsmooth (better results can be claimed if F is smooth)
- r is a regularization term (e.g., sparsity-promoting term in sparse neural network training [Scardapane et. al'17])

Example I: phase retrieval

To recover phase from a set of measured magnitude

$$\min_{\mathbf{x}} \frac{1}{m} \sum_{i=1}^{m} \left| |\mathcal{A}_i(\mathbf{x})|^2 - |b_i|^2 \right|$$



- $\mathbf{x}$ represents the underlying signal/image
- $\mathcal{A}_i$ the i-th measuring operation
- b_i the i-th measured magnitude

Remark: the objective is nonconvex nonsmooth

Picture from `https://aimi.stanford.edu/radiopaedia-list-ai-imaging-datasets`

Example II: sparse deep learning [Scardapane et al'17]

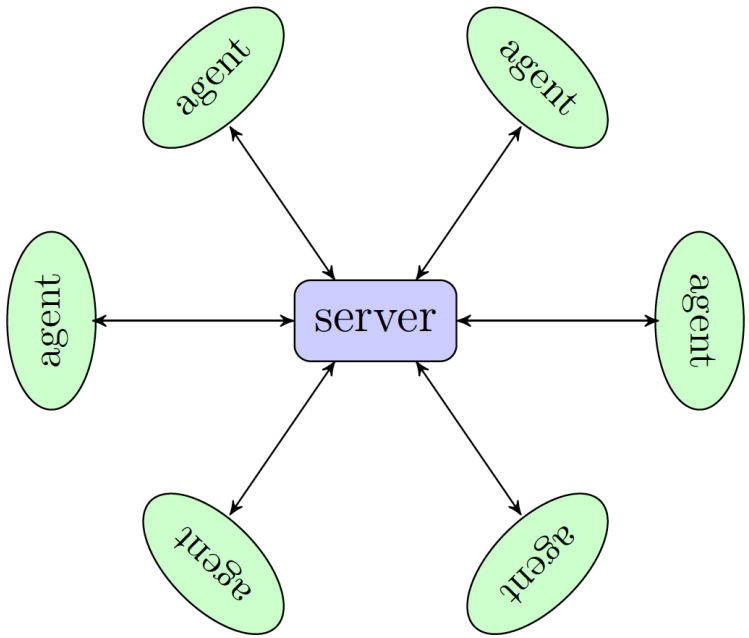
Sparsity-regularized deep learning model

$$\min_{\boldsymbol{\theta}} \frac{1}{m} \sum_{i=1}^{m} \ell(\boldsymbol{\theta}; \mathbf{x}_i, y_i) + r_1(\boldsymbol{\theta}) + r_{2,1}(\boldsymbol{\theta})$$

- $\boldsymbol{\theta}$: neural network parameters
- $(\mathbf{x}_i, y_i)$: training data point with label y_i
- r_1: ℓ_1-norm sparsity term
- $r_{2,1}$: group sparsity term (to remove all connections from one neuron)

Benefits: mitigating over-fitting, generating lighter model (that can reduce inference time and save storage)

Centralized computing architecture



- data distributed over working agents (or workers)
- model updated on server (or master)

parallel stochastic proximal subgradient method

Update by the parallel stochastic proximal subgradient method:

$$\mathbf{x}^{(k+1)} = \mathbf{prox}_{\alpha_k r}\left(\mathbf{x}^{(k)} - \frac{\alpha_k}{p}\sum_{j=1}^{p}\tilde{\nabla} f(\mathbf{x}^{(k)}; \xi_{k,j})\right)$$

- p: the number of workers
- $\tilde{\nabla} f$ denotes the subgradient
- $\xi_{k,j}$: the sample on j-th worker at the k-th update (can be a minibatch of samples)

Limitations:

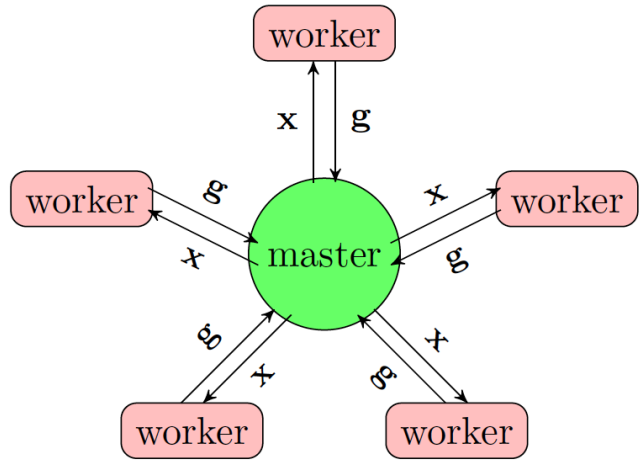
1. All workers are synchronized; this can cause waiting time and lead to low parallelization speed-up
2. Slow convergence

We address the limitations by asynchronous computing and acceleration

Proposed Algorithm

1. Initialize $\mathbf{x}^{(0)} \in \mathrm{dom}(r)$ and set $\mathbf{x}^{(1)} = \mathbf{x}^{(0)}$
2. Iterate for $k = 1, 2, \ldots$
   - Let $\mathbf{g}^{(k)} = \tilde{\nabla} f(\mathbf{x}^{(k-\tau_k)}; \xi_k)$ computed by a worker;
   - Choose $\alpha_k > 0$ and $\beta_k \geq 0$;
   - Update the variable $\mathbf{x}$ by

$$\mathbf{x}^{(k+1)} = \mathbf{prox}_{\alpha_k r}\left(\mathbf{x}^{(k)} - \alpha_k \mathbf{g}^{(k)} + \beta_k(\mathbf{x}^{(k)} - \mathbf{x}^{(k-1)})\right).$$

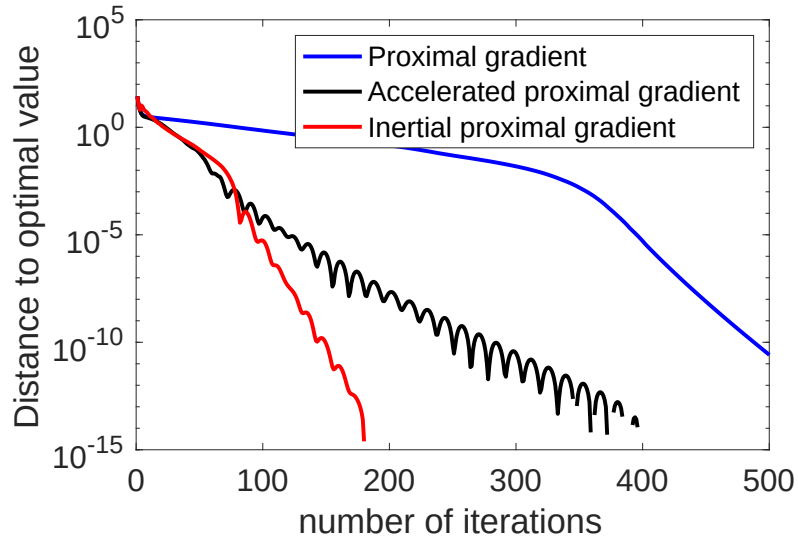


Remark

- update performed by master
- inertial term $\beta_k(\mathbf{x}^{(k)} - \mathbf{x}^{(k-1)})$ for acceleration
- ξ_k is a sample of ξ
- $\tau_k \geq 0$ measures the possible delay (i.e., asynchronous update to save communication time)

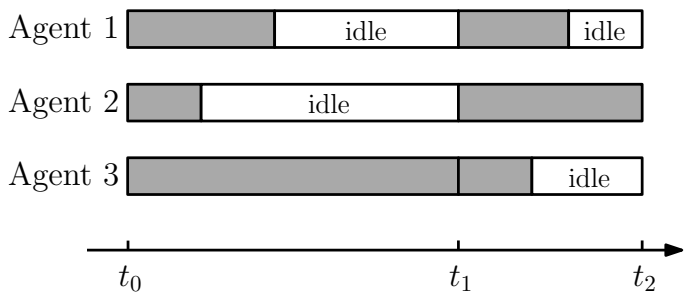
Heavy-ball or inertial acceleration

- first presented by Polyak'64
- closely related to Nesterov's acceleration, e.g.,
  $\mathbf{x}^{(k+1)} = \widehat{\mathbf{x}}^{(k)} - \alpha \cdot \mathrm{grad}(\widehat{\mathbf{x}}^{(k)})$ with $\widehat{\mathbf{x}}^{(k)} = \mathbf{x}^{(k)} + \omega_k(\mathbf{x}^{(k)} - \mathbf{x}^{(k-1)})$
- optimal convergence for strongly-convex quadratic programs but not for general convex programs
- nevertheless, often yields good numerical performance

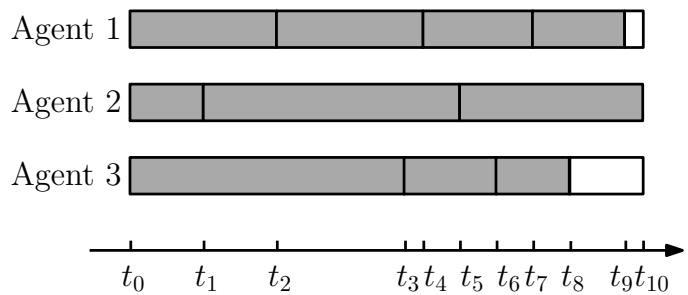


synchronous versus asynchronous computing

- asynchronous computing dates back as early as to [Bertsekas-Tsitsiklis'91]
- synchronization often causes idle waiting time



(a) synchronous



(b) asynchronous

Literature

Key ingredients of the proposed algorithm: *stochastic subgradient*, *inertial-acceleration*, *nonconvexity*, *asynchronous distributed*. Many existing works have some (but not all) of these ingredients.

- stochastic approximation or subgradient [Robbins-Monro'51; Nemirovski et. al'09; Davis-Drusvyatskiy'19; ...]
- heavy-ball or inertial-accelerated methods for convex or nonconvex problems [Polyak'64; Ochs et. al'14; Sun et. al'19; Loizou-Richtarik'20; ...]
- distributed stochastic (sub)gradient [Agarwal-Duchi'11; Recht et. al'11; Lian et. al'15; Sra et. al'16; Mai-Johansson'20; ...]

Theoretical results (informal)

Key assumption: the delay is upper bounded by τ (that is roughly #workers).
Let α be the learning rate and K the maximum number of iterations.

1. nondifferentiable but weakly-convex loss (i.e., f is nondifferentiable but $f + \frac{\rho}{2}\|\cdot\|^2$ is convex for some $\rho > 0$)

   $$\text{violation of optimality condition} = O\left(\frac{1}{\sqrt{K}}\left(C_0 + \frac{\alpha\tau^2}{\sqrt{K}} + \alpha^2\tau\right)\right)$$

   where C_0 is a constant for the no-delay case.

2. smooth loss with nondifferentiable regularization

   $$\text{violation of optimality condition} = O\left(\frac{1}{\sqrt{K}}\left(C_0 + \frac{\alpha\tau^2}{\sqrt{K}}\right)\right)$$

3. non-regularized smooth loss (i.e., f is smooth and $r \equiv 0$)

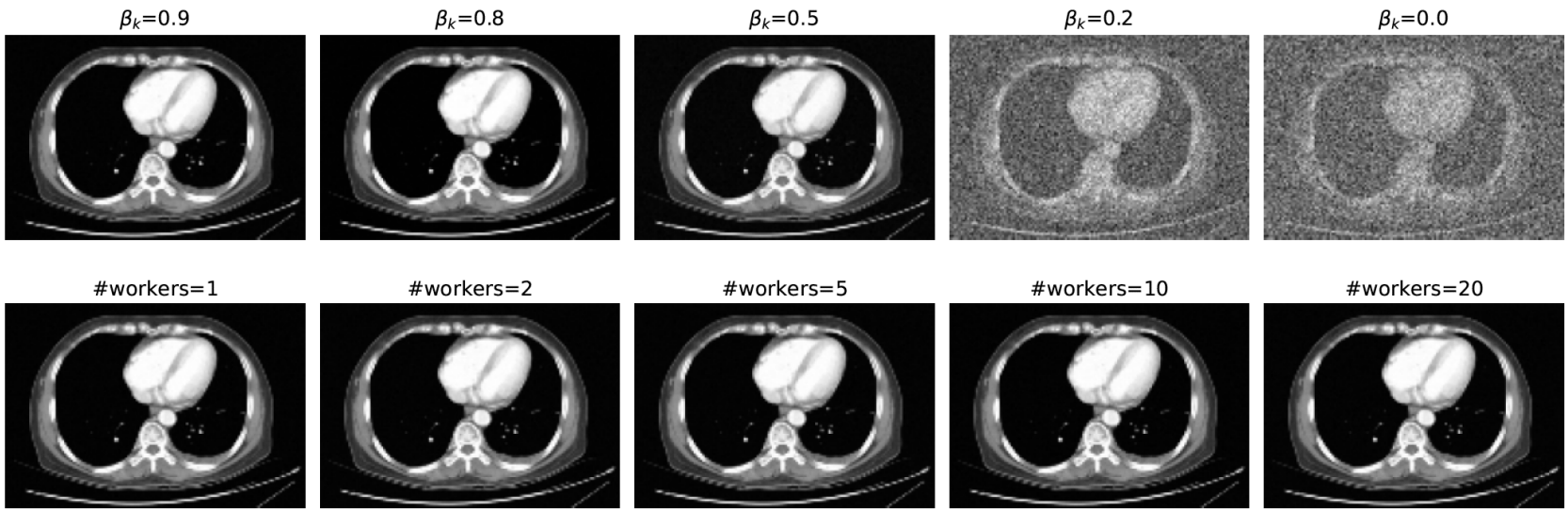
   $$\text{violation of optimality condition} = O\left(\frac{1}{\sqrt{K}}\left(C_0 + \frac{\alpha\tau}{\sqrt{K}}\right)\right)$$

Observation: the method can tolerate larger delay for nicer problems

Experiment I: phase retrieval

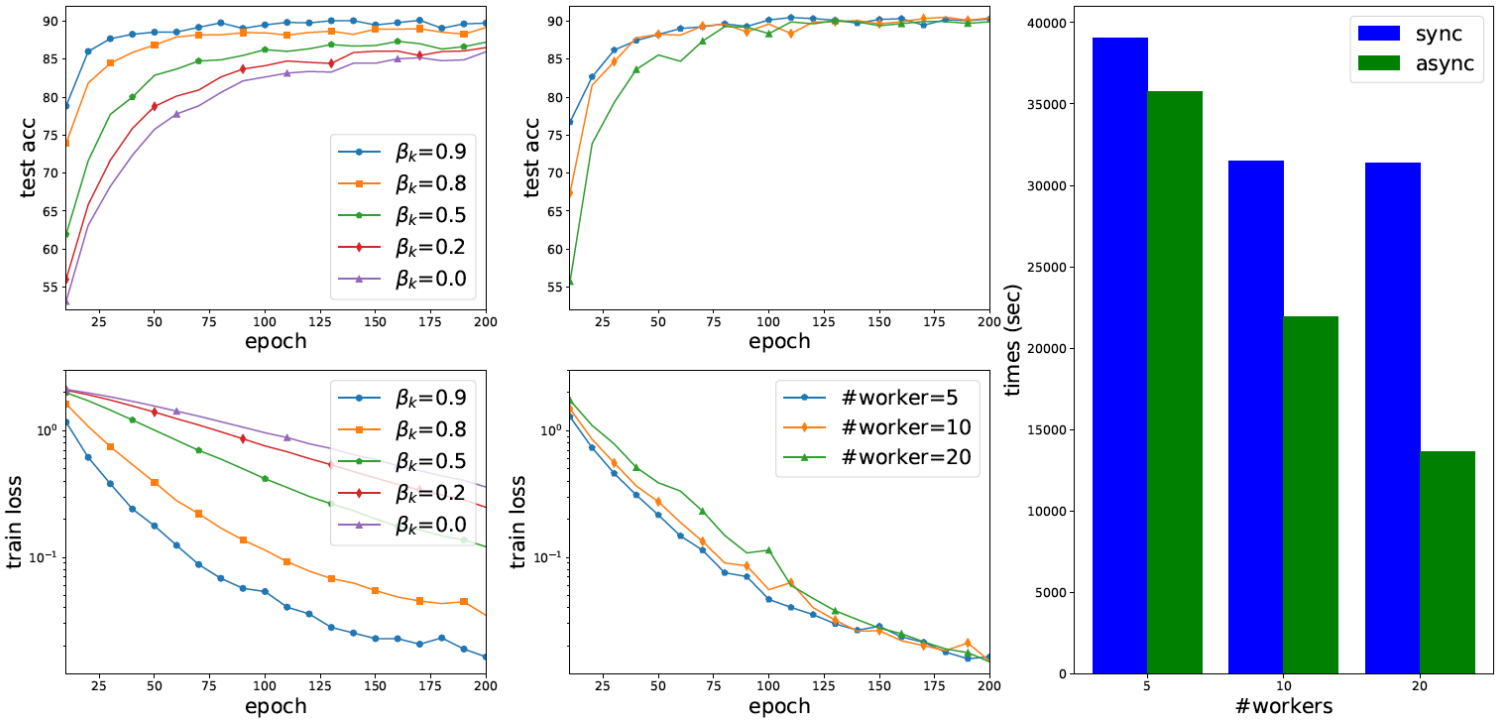
Apply the proposed algorithm to

$$\min_{\mathbf{x}} \frac{1}{m} \sum_{i=1}^{m} \left| |\mathcal{A}_i(\mathbf{x})|^2 - |b_i|^2 \right|$$



- for each setting, the algorithm runs to 400 epochs
- **observations:** inertial term accelerates convergence; delay almost does not affect convergence speed

Experiment II: training 9-layer AllCNN [Springenberg et. al] on cifar10



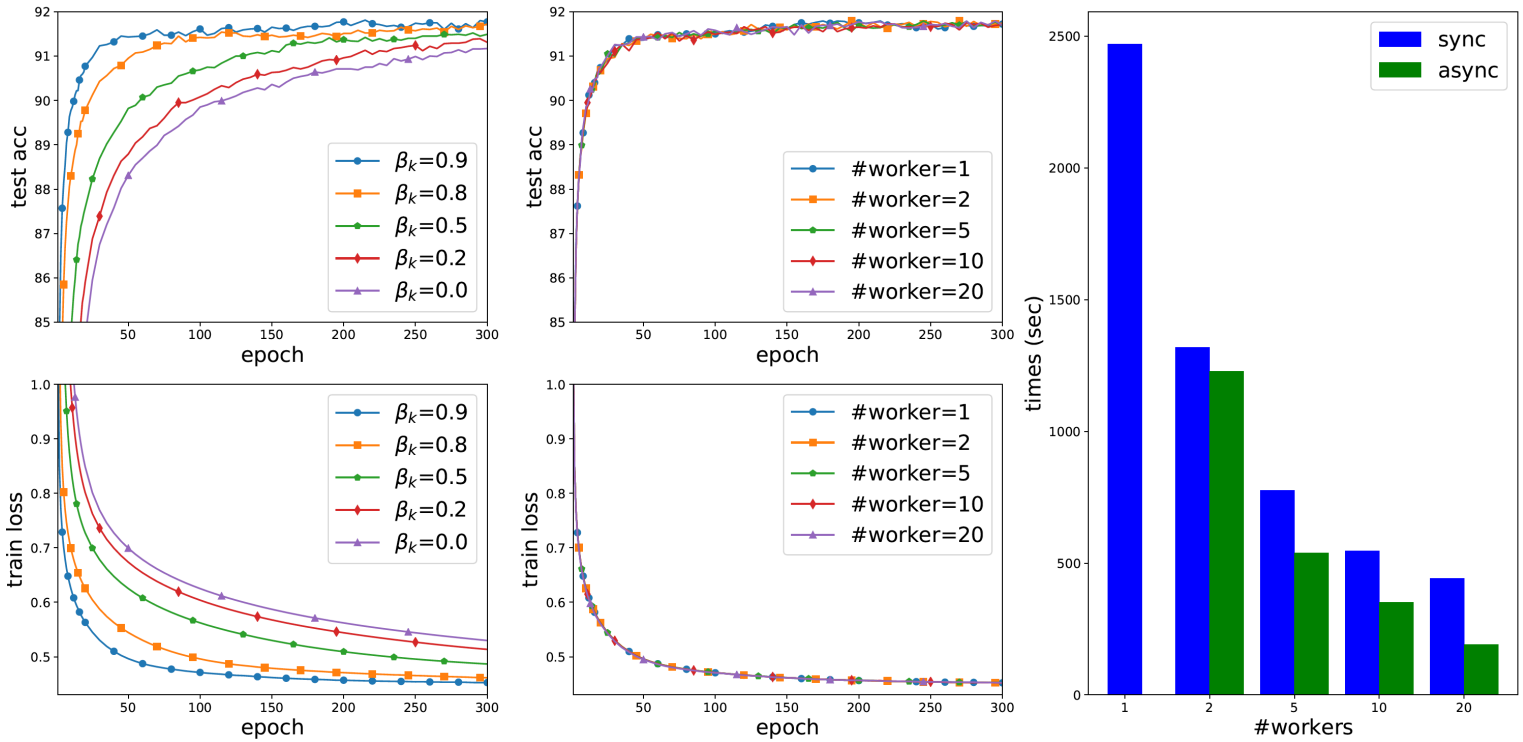
Observations

- inertial term accelerates convergence
- delay slightly affects the convergence speed
- asynchronous update significantly improves the parallelization performance

Experiment III: sparse bilinear logistic regression on MNIST

Apply the proposed proximal stochastic gradient method to

$$\min_{\mathcal{U},\mathcal{V},\mathbf{b}} -\frac{1}{m}\sum_{i=1}^{m}\log\left(\frac{\exp[\mathrm{tr}(U_{y_i}X_iV_{y_i})+b_{y_i}]}{\sum_{j=1}^{C}\exp[\mathrm{tr}(U_jX_iV_j)+b_j]}\right)+\lambda(\|\mathcal{U}\|_1+\|\mathcal{V}\|_1+\|\mathbf{b}\|_1),$$



Conclusions

- presented a distributed stochastic proximal (sub)gradient method for solving weakly-convex or smooth nonconvex stochastic problems
  - master-worker architecture adopted
  - outdated (sub)gradient allowed (due to asychrony)
- convergence rate results (in terms of stationarity violation) are given
  - with weak-convexity but not smoothness, delay has non-decaying effect on convergence rate
  - with smoothness, delay effect decays about the iteration number
- numerical results shown for phase retrieval and (deep) machine learning
  - inertial term accelerates the convergence
  - delay almost does not slow down the convergence
  - asynchronous computing yields higher parallelization speed up over the synchronous implementation

Reference

Yangyang Xu, Yibo Xu, Yonggui Yan, and Jie Chen. Distributed stochastic inertial-accelerated methods with delayed derivatives for nonconvex problems. To appear in *SIAM Journal on Imaging Sciences*, arXiv:2107.11513.

Reference

Yangyang Xu, Yibo Xu, Yonggui Yan, and Jie Chen. Distributed stochastic inertial-accelerated methods with delayed derivatives for nonconvex problems. To appear in *SIAM Journal on Imaging Sciences*, arXiv:2107.11513.

Thank you!!!