Balancing Communication and Computations in Gradient Tracking Methods for Distributed Optimization

Shagun Gupta

Raghu Bollapragada and Albert S. Berahas

Problem

$$\min_{x \in \mathbb{R}^p} f(x) = \sum_{i=1}^{n} f_i(x)$$

Each function f_i is only known to agent i $\forall$ $i = 1, 2, ..., n$

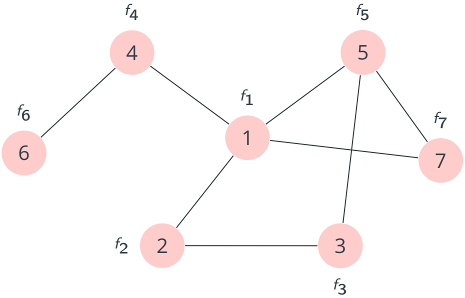
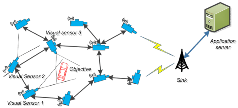


Figure: Distributed Network Example

Problem

$$\min_{x \in \mathbb{R}^p} f(x) = \sum_{i=1}^{n} f_i(x)$$

Each function f_i is only known to agent i $\forall$ $i = 1, 2, \ldots, n$



(a) Sensor Networks

You et al. 2013

(b) Machine Learning

Tom Taulli, Forbes 2019

(c) Signal Processing

Signal Processing, SINTEF 2022

Consensus Optimization Problem

$$\min_{x_i \in \mathbb{R}^p} \sum_{i=1}^{n} f_i(x_i)$$

$$s.t. \quad x_i = x_j \quad \forall\ i, j \in \mathcal{E}$$

Each node keeps a local copy $x_i \ \forall\ i = 1, 2, ..., n$

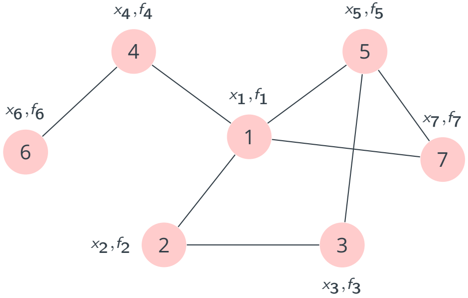


Figure: Distributed Network Example

Consensus Optimization Problem

$$\min_{x_i \in \mathbb{R}^p} f(\mathbf{x}) = \sum_{i=1}^{n} f_i(x_i)$$
$$s.t. \quad (W \otimes I_p)\mathbf{x} = \mathbf{x}$$

- $\mathbf{x}$ is a concatenation of all local x_i's
- **W** is a symmetric doubly-stochastic matrix that defines the connections in the network

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} \in \mathbb{R}^{np}, \quad W = \begin{bmatrix} w_{11} & w_{12} & \cdots & w_{1n} \\ w_{21} & w_{22} & \cdots & w_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ w_{n1} & w_{n2} & \cdots & w_{nn} \end{bmatrix} \in \mathbb{R}^{n \times n}$$

Consensus Optimization Problem

$$\min_{x_i \in \mathbb{R}^p} f(\mathbf{x}) = \sum_{i=1}^{n} f_i(x_i)$$

$$s.t. \quad \mathbf{Z}\,\mathbf{x} = \mathbf{x}$$

- $\mathbf{x}$ is a concatenation of all local x_i's
- **W** is a symmetric doubly-stochastic matrix that defines the connections in the network

$$\mathbf{Z} = W \otimes I_p \in \mathbb{R}^{np \times np}$$

Literature Review

1. Sublinearly Converging Methods

   DGD [Bertsekas, Tsitsiklis, et al. 1989, Nedic and Ozdaglar 2009, Sundhar Ram et al. 2010, Tsianos et al. 2012], **NN** [Mokhtari et al. 2017], **NEAR-DGD** [Berahas et al. 2018], ...

2. Linearly Converging Methods

   Push-pull [Pu, Shi, et al. 2020], **DIGing** [Nedic, Olshevsky, et al. 2017], **EXTRA** [Shi et al. 2015], **SONATA** [Sun et al. 2022], **NEXT** [Di Lorenzo and Scutari 2015], **Aug-DGM** [Xu et al. 2015], ...

3. Asynchronous Methods

   [Bertsekas, Tsitsiklis, et al. 1989], [Ram, Veeravalli, and Nedic 2009], **HOGWILD** [Recht et al. 2011], [Wei and Ozdaglar 2013] ...

4. Stochastic Algorithms

   DSGT and **GSGT** [Pu and Nedić 2021], **ProxiSkip** [Mishchenko et al. 2022], ...

Distributed Gradient Descent (DGD)

DGD [Bertsekas, Tsitsiklis, et al. 1989, Nedic and Ozdaglar 2009, Sundhar Ram et al. 2010, Tsianos et al. 2012]

$$\mathbf{x}_{k+1} = \underbrace{\mathbf{Z}\,\mathbf{x}_k}_{\textit{Communication}} - \alpha \underbrace{\nabla \mathbf{f}(\mathbf{x}_k)}_{\textit{Computation}}$$

$$\mathbf{x}_k = \begin{bmatrix} x_{1,k} \\ x_{2,k} \\ \vdots \\ x_{n,k} \end{bmatrix} \in \mathbb{R}^{np}, \quad \nabla \mathbf{f}(\mathbf{x}_k) = \begin{bmatrix} \nabla f_1(x_{1,k}) \\ \nabla f_2(x_{2,k}) \\ \vdots \\ \nabla f_n(x_{n,k}) \end{bmatrix} \in \mathbb{R}^{np}$$

- Constant (α) : Linear Convergence to neighbourhood $\mathcal{O}(\alpha)$
- Diminishing (α) : Sublinear convergence to solution

Distributed Gradient Descent (DGD)

DGD [Bertsekas, Tsitsiklis, et al. 1989, Nedic and Ozdaglar 2009, Sundhar Ram et al. 2010, Tsianos et al. 2012]

$$\mathbf{x}_{k+1} = \underbrace{\mathbf{Z}\mathbf{x}_k}_{\textit{Communication}} - \alpha \underbrace{\nabla \mathbf{f}(\mathbf{x}_k)}_{\textit{Computation}}$$

[Berahas et al. 2018] proposed a variant of DGD where increasing communications achieves linear convergence under constant α.

Gradient Tracking Algorithms

Push-pull [Pu, Shi, et al. 2020], **DIGing** [Nedic, Olshevsky, et al. 2017], **EXTRA** [Shi et al. 2015], **SONATA** [Sun et al. 2022], **NEXT** [Di Lorenzo and Scutari 2015], **Aug-DGM** [Xu et al. 2015], ...

$$\mathbf{x}_{k+1} = \mathbf{Z}\mathbf{x}_k - \alpha\mathbf{y}_k,$$
$$\mathbf{y}_{k+1} = \mathbf{Z}\mathbf{y}_k + \nabla f(\mathbf{x}_{k+1}) - \nabla f(\mathbf{x}_k)$$

$$\mathbf{x}_k = \begin{bmatrix} x_{1,k} \\ x_{2,k} \\ \vdots \\ x_{n,k} \end{bmatrix} \in \mathbb{R}^{np}, \quad \mathbf{y}_k = \begin{bmatrix} y_{1,k} \\ y_{2,k} \\ \vdots \\ y_{n,k} \end{bmatrix} \in \mathbb{R}^{np}, \quad \nabla \mathbf{f}(\mathbf{x}_k) = \begin{bmatrix} \nabla f_1(x_{1,k}) \\ \nabla f_2(x_{2,k}) \\ \vdots \\ \nabla f_n(x_{n,k}) \end{bmatrix} \in \mathbb{R}^{np}$$

- Use an additional dual variable $\mathbf{y}_k$ to track the gradient
- Constant α : Linear converge to solution

This Talk

1. We develop generalised gradient tracking algorithms for distributed optimization with flexibility in:

   - Communication Structure
   - Multiple Communication Steps
   - Multiple Computation Steps

2. Provide convergence conditions for each level of flexibility.

3. Illustrate benefits of these methods with numerical analysis.

Base Algorithm

Single communication and computation steps in each iteration.

$$\mathbf{x}_{k+1} = \mathbf{Z}_1\mathbf{x}_k - \alpha\mathbf{Z}_2\mathbf{y}_k,$$
$$\mathbf{y}_{k+1} = \mathbf{Z}_3\mathbf{y}_k + \mathbf{Z}_4(\nabla\mathbf{f}(\mathbf{x}_{k+1}) - \nabla\mathbf{f}(\mathbf{x}_k))$$

- Each mixing matrix W_1, W_2, W_3 and W_4 is symmetric and doubly stochastic.
- $\beta_1, \beta_2, \beta_3$ and β_4 are the corresponding 2nd highest eigenvalues.

Base Algorithm

Definitions

$$\bar{x}_k = \frac{1}{n}\sum_{i=1}^{n} x_{i,k}, \quad \bar{y}_k = \frac{1}{n}\sum_{i=1}^{n} y_{i,k}$$

Base Algorithm

Definitions

$$\bar{x}_k = \frac{1}{n}\sum_{i=1}^{n} x_{i,k}, \quad \bar{y}_k = \frac{1}{n}\sum_{i=1}^{n} y_{i,k}$$

$$\bar{\mathbf{x}}_k = \begin{bmatrix} \bar{x}_k \\ \bar{x}_k \\ \vdots \\ \bar{x}_k \end{bmatrix} \in \mathbb{R}^{np}, \quad \bar{\mathbf{y}}_k = \begin{bmatrix} \bar{y}_k \\ \bar{y}_k \\ \vdots \\ \bar{y}_k \end{bmatrix} \in \mathbb{R}^{np}, \quad r_k = \begin{bmatrix} \|\bar{x}_k - x^*\|_2 \\ \|\mathbf{x}_k - \bar{\mathbf{x}}_k\|_2 \\ \|\mathbf{y}_k - \bar{\mathbf{y}}_k\|_2 \end{bmatrix}$$

Base Algorithm

Assumption

1. Each component function f_i is $\mu > 0$ strongly convex and has $L > 0$ Lipschitz continuous gradients.

Base Algorithm

Assumption

1. Each component function f_i is $\mu > 0$ strongly convex and has $L > 0$ Lipschitz continuous gradients.

Theorem

Suppose Assumption 1 holds and $\beta_1, \beta_3 < 1$ in Base Algorithm, then $\exists\ \alpha > 0$, s.t. $\|r_k\|_2$ goes to 0 at a linear rate.

$$\mathbf{x}_{k+1} = \mathbf{Z}_1 \mathbf{x}_k - \alpha \mathbf{Z}_2 \mathbf{y}_k,$$
$$\mathbf{y}_{k+1} = \mathbf{Z}_3 \mathbf{y}_k + \mathbf{Z}_4 (\nabla \mathbf{f}(\mathbf{x}_{k+1}) - \nabla \mathbf{f}(\mathbf{x}_k))$$

Base Algorithm Cases

The mixing matrix W has $\beta < 1$, $\quad \mathbf{Z} = W \otimes I_p$

GTM_1 (DIGing, EXTRA, ...)

$$\mathbf{x}_{k+1} = \mathbf{Z}\mathbf{x}_k - \alpha \mathbf{y}_k$$
$$\mathbf{y}_{k+1} = \mathbf{Z}\mathbf{y}_k + \nabla \mathbf{f}(\mathbf{x}_{k+1}) - \nabla \mathbf{f}(\mathbf{x}_k)$$

GTM_2 (NEXT, SONATA, ...)

$$\mathbf{x}_{k+1} = \mathbf{Z}\left(\mathbf{x}_k - \alpha \mathbf{y}_k\right)$$
$$\mathbf{y}_{k+1} = \mathbf{Z}\mathbf{y}_k + \nabla \mathbf{f}(\mathbf{x}_{k+1}) - \nabla \mathbf{f}(\mathbf{x}_k)$$

GTM_3 (Aug-DMM, ATC-DIGing, ...)

$$\mathbf{x}_{k+1} = \mathbf{Z}\left(\mathbf{x}_k - \alpha \mathbf{y}_k\right)$$
$$\mathbf{y}_{k+1} = \mathbf{Z}\left(\mathbf{y}_k + \nabla \mathbf{f}(\mathbf{x}_{k+1}) - \nabla \mathbf{f}(\mathbf{x}_k)\right)$$

Base Algorithm Cases - Rate of Convergence

Theorem

Suppose Assumption 1 holds and $\alpha \leq \frac{1}{L}$, $\|r_k\|_2$ *goes to 0 at a linear rate upper bounded by the following expressions*

For GTM_1 $\quad \max\left\{1 - \frac{\alpha\mu}{2},\ \beta + \sqrt{\alpha L}\left(2.5 + \sqrt{\kappa}\right)\right\}$

For GTM_2 $\quad \max\left\{1 - \frac{\alpha\mu}{2},\ \beta + \sqrt{\alpha L}\left(2.5 + \sqrt{\kappa\beta}\right)\right\}$

For GTM_3 $\quad \max\left\{1 - \frac{\alpha\mu}{2},\ \beta\left(1 + \sqrt{\alpha L}\left(2.5 + \sqrt{\kappa}\right)\right)\right\}$

where $\kappa = \frac{L}{\mu}$.

Base Algorithm Cases - Numerical Experiments

Almost Full network $\beta = 0.25$

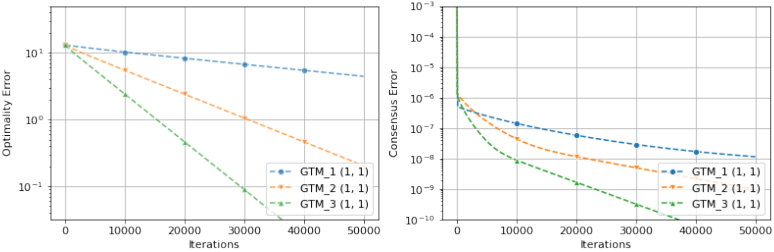


Figure: Quadratics, $n = 16$, $p = 10$, $\kappa = 10^4$

Base Algorithm Cases - Numerical Experiments

Cyclic Network $\beta = 0.992$

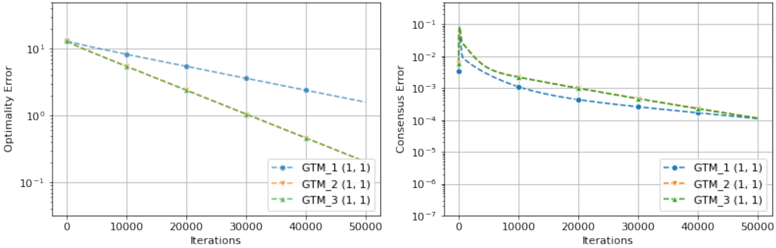
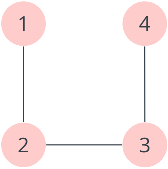


Figure: Quadratics, $n = 16$, $p = 10$, $\kappa = 10^4$

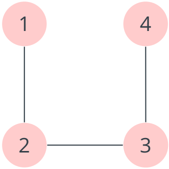
Multiple Communications



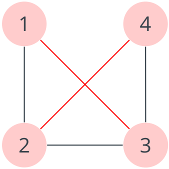
(a) Single Communication

$$W = \begin{bmatrix} 0.9 & 0.1 & 0 & 0 \\ 0.1 & 0.8 & 0.1 & 0 \\ 0 & 0.1 & 0.8 & 0.1 \\ 0 & 0 & 0.1 & 0.9 \end{bmatrix}$$

Multiple Communications



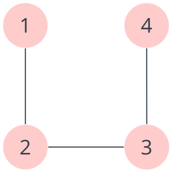
(a) Single Communication



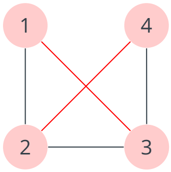
(b) 2 Communications

$$W = \begin{bmatrix} 0.9 & 0.1 & 0 & 0 \\ 0.1 & 0.8 & 0.1 & 0 \\ 0 & 0.1 & 0.8 & 0.1 \\ 0 & 0 & 0.1 & 0.9 \end{bmatrix}$$

Multiple Communications



(a) Single Communication

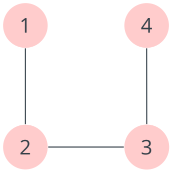


(b) 2 Communications

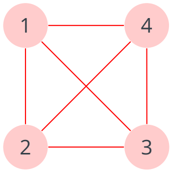
$$W = \begin{bmatrix} 0.9 & 0.1 & 0 & 0 \\ 0.1 & 0.8 & 0.1 & 0 \\ 0 & 0.1 & 0.8 & 0.1 \\ 0 & 0 & 0.1 & 0.9 \end{bmatrix}$$

$$W^{2} = \begin{bmatrix} 0.82 & 0.17 & 0.01 & 0 \\ 0.17 & 0.66 & 0.16 & 0.01 \\ 0.01 & 0.16 & 0.66 & 0.17 \\ 0 & 0.01 & 0.17 & 0.82 \end{bmatrix}$$

Multiple Communications

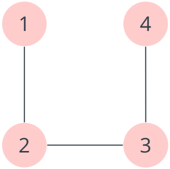


(a) Single Communication

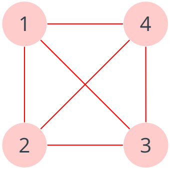


(b) 200 Communications

Multiple Communications



(a) Single Communication



(b) 200 Communications

$$W = \begin{bmatrix} 0.9 & 0.1 & 0 & 0 \\ 0.1 & 0.8 & 0.1 & 0 \\ 0 & 0.1 & 0.8 & 0.1 \\ 0 & 0 & 0.1 & 0.9 \end{bmatrix} \qquad W^{200} = \begin{bmatrix} 0.25 & 0.25 & 0.25 & 0.25 \\ 0.25 & 0.25 & 0.25 & 0.25 \\ 0.25 & 0.25 & 0.25 & 0.25 \\ 0.25 & 0.25 & 0.25 & 0.25 \end{bmatrix}$$

Multiple Communications

$t \rightarrow \#$ of communication steps

$$\mathbf{x}_{k+1} = \mathbf{Z}_1^t \mathbf{x}_k - \alpha \mathbf{Z}_2^t \mathbf{y}_k,$$
$$\mathbf{y}_{k+1} = \mathbf{Z}_3^t \mathbf{y}_k + \mathbf{Z}_4^t (\nabla \mathbf{f}(\mathbf{x}_{k+1}) - \nabla \mathbf{f}(\mathbf{x}_k))$$

Multiple Communications Cases

GTM_1

$$\mathbf{x}_{k+1} = \mathbf{Z}^t \mathbf{x}_k - \alpha \mathbf{y}_k$$
$$\mathbf{y}_{k+1} = \mathbf{Z}^t \mathbf{y}_k + \nabla \mathbf{f}(\mathbf{x}_{k+1}) - \nabla \mathbf{f}(\mathbf{x}_k)$$

GTM_2

$$\mathbf{x}_{k+1} = \mathbf{Z}^t \left(\mathbf{x}_k - \alpha \mathbf{y}_k\right)$$
$$\mathbf{y}_{k+1} = \mathbf{Z}^t \mathbf{y}_k + \nabla \mathbf{f}(\mathbf{x}_{k+1}) - \nabla \mathbf{f}(\mathbf{x}_k)$$

GTM_3

$$\mathbf{x}_{k+1} = \mathbf{Z}^t \left(\mathbf{x}_k - \alpha \mathbf{y}_k\right)$$
$$\mathbf{y}_{k+1} = \mathbf{Z}^t \left(\mathbf{y}_k + \nabla \mathbf{f}(\mathbf{x}_{k+1}) - \nabla \mathbf{f}(\mathbf{x}_k)\right)$$

Multiple Communications - Rate of Convergence

Theorem

Suppose Assumption 1 holds, number of communications is atleast 1 ($t \geq 1$) and $\alpha \leq \frac{1}{L}$, $\|r_k\|_2$ goes to 0 at a linear rate upper bounded by the following expressions

For GTM_1 $\quad \max\left\{1 - \frac{\alpha\mu}{2},\ \beta^t + \sqrt{\alpha L}\left(2.5 + \sqrt{\kappa}\right)\right\}$

For GTM_2 $\quad \max\left\{1 - \frac{\alpha\mu}{2},\ \beta^t + \sqrt{\alpha L}\left(2.5 + \sqrt{\kappa\beta^t}\right)\right\}$

For GTM_3 $\quad \max\left\{1 - \frac{\alpha\mu}{2},\ \beta^t\left(1 + \sqrt{\alpha L}\left(2.5 + \sqrt{\kappa}\right)\right)\right\}$

where $\kappa = \frac{L}{\mu}$.

Multiple Communications - Numerical Experiments

Cyclic Network $\beta = 0.992$

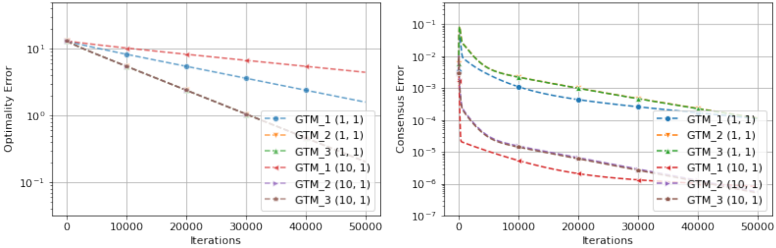


Figure: Quadratics, $n = 16$, $p = 10$, $\kappa = 10^4$

Multiple Communications and Computations

$t \rightarrow \#$ of communication steps
$g \rightarrow \#$ of computation steps

$$\mathbf{u}_1 = \mathbf{x}_k$$
$$\mathbf{v}_1 = \mathbf{y}_k$$

Multiple Communications and Computations

$t \rightarrow \#$ of communication steps
$g \rightarrow \#$ of computation steps

$$\mathbf{u}_1 = \mathbf{x}_k$$
$$\mathbf{v}_1 = \mathbf{y}_k$$

For $i \rightarrow 1, 2, ..., g-1$

$$\mathbf{u}_{i+1} = \mathbf{u}_i - \alpha \mathbf{v}_i,$$
$$\mathbf{v}_{i+1} = \mathbf{v}_i + \nabla \mathbf{f}(\mathbf{u}_{i+1}) - \nabla \mathbf{f}(\mathbf{u}_u)$$

Multiple Communications and Computations

$t \rightarrow \#$ of communication steps
$g \rightarrow \#$ of computation steps

$$\mathbf{u}_1 = \mathbf{x}_k$$
$$\mathbf{v}_1 = \mathbf{y}_k$$
For $i \rightarrow 1, 2, ..., g-1$
$$\mathbf{u}_{i+1} = \mathbf{u}_i - \alpha \mathbf{v}_i,$$
$$\mathbf{v}_{i+1} = \mathbf{v}_i + \nabla \mathbf{f}(\mathbf{u}_{i+1}) - \nabla \mathbf{f}(\mathbf{u}_u)$$
$$\mathbf{x}_{k+1} = \mathbf{Z}_1^t \mathbf{u}_g - \alpha \mathbf{Z}_2^t \mathbf{v}_g,$$
$$\mathbf{y}_{k+1} = \mathbf{Z}_3^t \mathbf{v}_g + \mathbf{Z}_4^t (\nabla \mathbf{f}(\mathbf{x}_{k+1}) - \nabla \mathbf{f}(\mathbf{v}_g))$$

Multiple Communications and Computations

$t \rightarrow \#$ of communication steps
$g \rightarrow \#$ of computation steps

Theorem

Suppose Assumption 1 holds, $\beta_1, \beta_3 < 1$, number of communication steps is at least one ($t \geq 1$) and number of computation steps is finite ($g < \infty$), then $\exists\ \alpha > 0$, s.t. $\|r_k\|_2$ goes to 0 at a linear rate.

Multiple Communications and Computations

$t \rightarrow \#$ of communication steps
$g \rightarrow \#$ of computation steps

Theorem

Suppose Assumption 1 holds, $\beta_1, \beta_3 < 1$, number of communication steps is at least one ($t \geq 1$) and number of computation steps is finite ($g < \infty$), then $\exists\ \alpha > 0$, s.t. $\|r_k\|_2$ goes to 0 at a linear rate.

Quantifying effect of multiple computations steps is ongoing work.

Multiple Communications and Computations Cases

GTM_1

$$\mathbf{x}_{k+1} = \mathbf{Z}^t \mathbf{u}_g - \alpha \mathbf{v}_g$$
$$\mathbf{y}_{k+1} = \mathbf{Z}^t \mathbf{v}_g + \nabla \mathbf{f}(\mathbf{x}_{k+1}) - \nabla \mathbf{f}(\mathbf{u}_g)$$
$\rightarrow g - 1$ compute steps

GTM_2

$$\mathbf{x}_{k+1} = \mathbf{Z}^t (\mathbf{u}_g - \alpha \mathbf{v}_g)$$
$$\mathbf{y}_{k+1} = \mathbf{Z}^t \mathbf{v}_g + \nabla \mathbf{f}(\mathbf{x}_{k+1}) - \nabla \mathbf{f}(\mathbf{u}_g)$$
$\rightarrow g - 1$ compute steps

GTM_3

$$\mathbf{x}_{k+1} = \mathbf{Z}^t (\mathbf{u}_g - \alpha \mathbf{v}_g)$$
$$\mathbf{y}_{k+1} = \mathbf{Z}^t (\mathbf{v}_g + \nabla \mathbf{f}(\mathbf{x}_{k+1}) - \nabla \mathbf{f}(\mathbf{u}_g))$$
$\rightarrow g - 1$ compute steps

Multiple Communications and Computations - Numerical Experiments

Cyclic Network $\beta = 0.992$

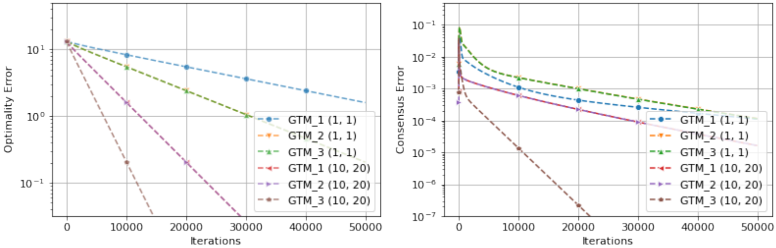


Figure: Quadratics, $n = 16$, $p = 10$, $\kappa = 10^4$

Conclusions

1. We propose generalised gradient tracking algorithms that provide flexibility with respect to communication structure, communication and computation overhead.

2. Adapting your algorithm to the system with this flexibility can allow you to improve convergence rate.

Thank You!

Questions?

Backup Slides