Balancing Communications and Computations in Gradient Tracking Methods for Decentralized Optimization

Shagun Gupta,
Raghu Bollapragada and Albert S. Berahas

Problems

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

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

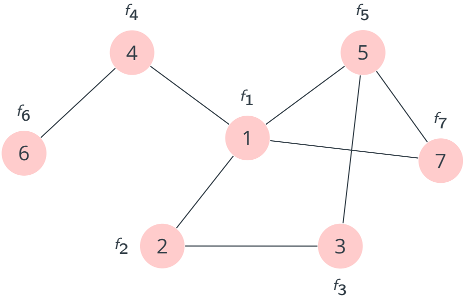
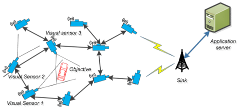


Figure: Distributed Network Example

Problems

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

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

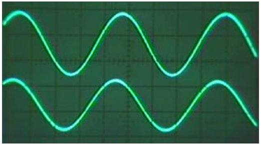


(a) Sensor Networks

You et al. 2013

(b) Machine Learning

Tom Taulli, Forbes 2019



(c) Signal Processing

Signal Processing, MIT OCW 2011

Consensus Optimization Problem

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

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

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

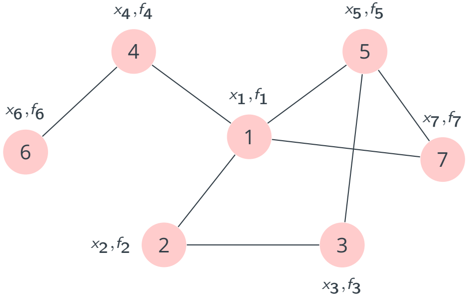


Figure: Distributed Network Example

Consensus Optimization Problem

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

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

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

Consensus Optimization Problem

$$\min_{x_i \in \mathbb{R}^d} f(\mathbf{x}) = \sum_{i=1}^{n} f_i(x_i)$$
$$s.t. \quad \mathbf{Z}\,\mathbf{x} = \mathbf{x}$$

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

$$\mathbf{Z} = \mathbf{W} \otimes I_d \in \mathbb{R}^{nd \times nd}$$

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], **LU-GT** [Nguyen et al. 2022], ...

3. Asynchronous Methods

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

4. Randomized Algorithms

   DSGT and **GSGT** [Pu and Nedić 2021], **ProxiSkip** [Mishchenko et al. 2022], **FedAvg** [McMahan et al. 2017], **FedLin** [Mitra et al. 2021], **Scaffold** [Karimireddy et al. 2020], **Scaffnew** [Mishchenko et al. 2022], ...

Gradient Tracking 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], ...

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

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

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

Gradient Tracking 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], ...

DIGing:

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

Aug-DGM:

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

Gradient Tracking 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], ...

DIGing:

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

Aug-DGM:

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

- Choice of information shared affects both convergence and practical implementation

Gradient Tracking 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], ...

DIGing:

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

Aug-DGM:

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

- Choice of information shared affects both convergence and practical implementation
- Applications require a different composition of communication and computation steps to achieve overall efficiency

This Talk

1. We develop a gradient tracking algorithmic framework ($\texttt{GTA}$) to unify gradient tracking methods.

2. Provide flexibility in number of communication and computation steps in each iteration with a:
   2.1 **Deterministic** scheme $\rightarrow$ $\texttt{GTA}$
   2.2 **Randomized** scheme $\rightarrow$ $\texttt{RGTA}$

3. Provide sufficient conditions for linear rate of convergence.

4. Illustrate benefits of this flexibility with numerical experiments.

GTA Framework

$\mathbf{W} \in \mathbb{R}^{n \times n} \rightarrow$ mixing matrix

- Symmetric, Doubly Stochastic
- Represents the network, i.e., $w_{ii} > 0$ and $w_{ij} > 0$ iff $(i, j) \in \mathcal{E}$
- $\left\| \mathbf{W} - \frac{1_n 1_n^T}{n} \right\|_2 = \beta \in [0, 1)$

GTA Framework

$\mathbf{W} \in \mathbb{R}^{n \times n} \rightarrow$ mixing matrix

- Symmetric, Doubly Stochastic
- Represents the network, i.e., $w_{ii} > 0$ and $w_{ij} > 0$ iff $(i, j) \in \mathcal{E}$
- $\left\| \mathbf{W} - \frac{1_n 1_n^T}{n} \right\|_2 = \beta \in [0, 1)$

$\mathbf{W}_1, \mathbf{W}_2, \mathbf{W}_3, \mathbf{W}_4 \in \mathbb{R}^{n \times n} \rightarrow$ communication matrices

- Symmetric, Doubly Stochastic
- Represents a subset of edges of the network, i.e., $w_{1,ii} > 0$ and $w_{1,ij} \geq 0$ if $(i, j) \in \mathcal{E}$ else $w_{1,ij} = 0$
- $\left\| \mathbf{W}_i - \frac{1_n 1_n^T}{n} \right\|_2 = \beta_i \in [0, 1] \quad \forall \quad i = 1, 2, 3, 4$

GTA Framework

$\mathbf{W}_1, \mathbf{W}_2, \mathbf{W}_3, \mathbf{W}_4 \rightarrow$ communication matrices

Single communication and computation step in each iteration.

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

where $\mathbf{Z}_i = \mathbf{W}_i \otimes I_d \in \mathbb{R}^{nd \times nd} \quad \forall \quad i = 1, 2, 3, 4$

GTA Framework Special Cases

Mixing matrix $\mathbf{W}$ and $\mathbf{Z} = \mathbf{W} \otimes I_d$

GTA-1 (DIGing, EXTRA, ...)

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

GTA-2 (NEXT, SONATA, ...)

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

GTA-3 (Aug-DGM, ATC-DIGing, ...)

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

GTA Framework - Step size condition

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

Assumption

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

Theorem

Suppose Assumption 1 holds, $\beta_1, \beta_3 < 1$ in GTA Framework and

$$\alpha < \min \left\{ \frac{1}{L}, \frac{1-\beta_3}{L\beta_4}, \frac{(1-\beta_1+2\beta_2)\mu}{2\beta_2 L(L+\mu)} \left(\sqrt{1 + \frac{4(1-\beta_1)(1-\beta_3)\beta_2(L+\mu)}{\mu\beta_4(1-\beta_1+2\beta_2)^2}} - 1 \right) \right\},$$

the iterates $\{x_k, y_k\}$ converge to the solution at a linear rate.

GTA Framework Cases - Rate of Convergence

Theorem

Suppose Assumption 1 holds and $\alpha \leq \frac{1}{L}$, iterates $\{x_k, y_k\}$ converge to the solution at a linear rate upper bounded by the following expressions

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

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

GTA-3: $$\max\left\{1 - \frac{\alpha\mu}{2},\ \beta\left(1 + \sqrt{\alpha L}\left(2.5 + \sqrt{\kappa}\right)\right)\right\}$$

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

GTA Framework - Numerical Experiments

Almost Full network $\beta = 0.25$

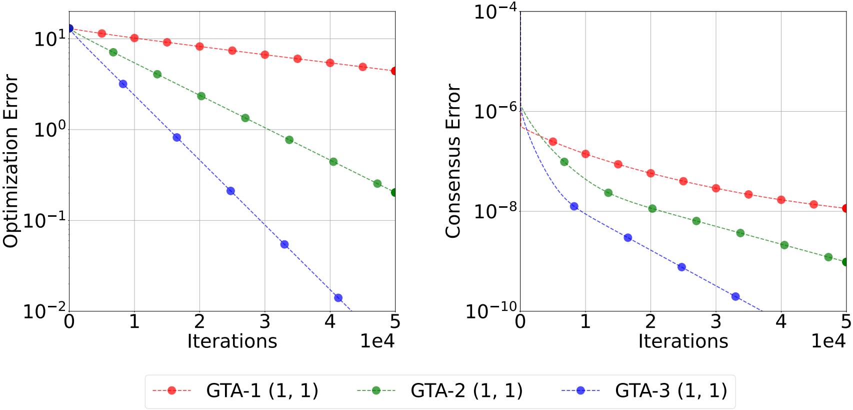


Figure: Quadratics, $n = 16$, $d = 10$, $\kappa = 10^4$

GTA Framework - Numerical Experiments

Cyclic Network $\beta = 0.992$

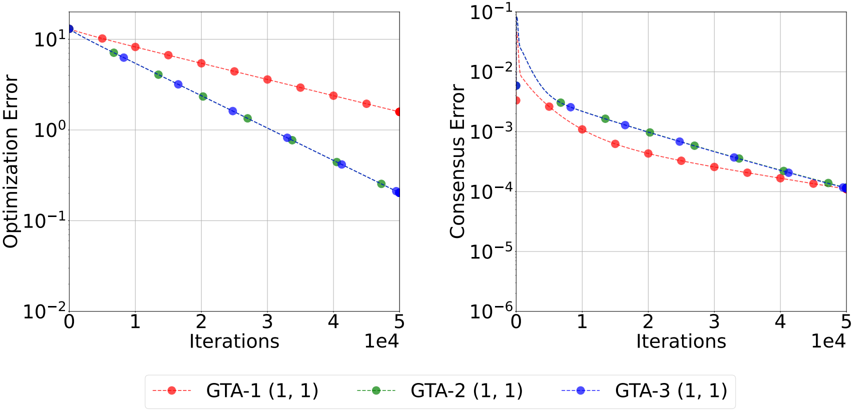
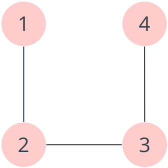


Figure: Quadratics, $n = 16$, $d = 10$, $\kappa = 10^4$

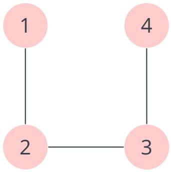
Multiple Communications



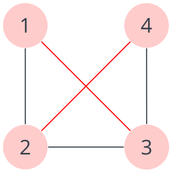
(a) Single Communication

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

Multiple Communications



(a) Single Communication



(b) 2 Communications

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

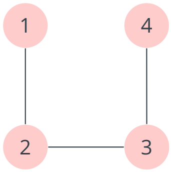
Multiple Communications

(a) Single Communication (b) 2 Communications

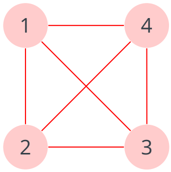
$$\mathbf{W} = \begin{bmatrix} 0.9 & 0.1 & 0 & 0 \\ 0.1 & 0.8 & 0.1 & 0 \\ 0 & 0.1 & 0.8 & 0.1 \\ 0 & 0 & 0.1 & 0.9 \end{bmatrix}$$

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

Multiple Communications

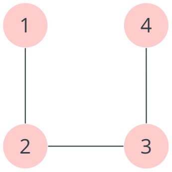


(a) Single Communication

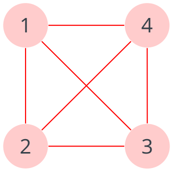


(b) 200 Communications

Multiple Communications



(a) Single Communication



(b) 200 Communications

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

$$\mathbf{W}^{200} = \begin{bmatrix} 0.25 & 0.25 & 0.25 & 0.25 \\ 0.25 & 0.25 & 0.25 & 0.25 \\ 0.25 & 0.25 & 0.25 & 0.25 \\ 0.25 & 0.25 & 0.25 & 0.25 \end{bmatrix}$$

GTA Framework - Multiple Communications

$n_c \rightarrow$ #of communication steps

$$\mathbf{W}_i \rightarrow \mathbf{W}_i^{n_c} \qquad \forall i = 1, 2, 3, 4$$

$$\beta_i \rightarrow \beta_i^{n_c} \qquad \forall i = 1, 2, 3, 4$$

$$\mathbf{Z}_i \rightarrow \mathbf{Z}_i^{n_c} = \mathbf{W}_i^{n_c} \otimes I_d \qquad \forall i = 1, 2, 3, 4$$

$$\mathbf{x}_{k+1} = \mathbf{Z}_1^{n_c}\mathbf{x}_k - \alpha\mathbf{Z}_2^{n_c}\mathbf{y}_k,$$

$$\mathbf{y}_{k+1} = \mathbf{Z}_3^{n_c}\mathbf{y}_k + \mathbf{Z}_4^{n_c}(\nabla\mathbf{f}(\mathbf{x}_{k+1}) - \nabla\mathbf{f}(\mathbf{x}_k))$$

GTA Framework - Multiple Communications

$n_c \rightarrow$ #of communication steps

$\mathbf{W}_i \rightarrow \mathbf{W}_i^{n_c} \qquad \forall i = 1, 2, 3, 4$

$\beta_i \rightarrow \beta_i^{n_c} \qquad \forall i = 1, 2, 3, 4$

$\mathbf{Z}_i \rightarrow \mathbf{Z}_i^{n_c} = \mathbf{W}_i^{n_c} \otimes I_d \qquad \forall i = 1, 2, 3, 4$

$$\mathbf{x}_{k+1} = \mathbf{Z}_1^{n_c}\mathbf{x}_k - \alpha\mathbf{Z}_2^{n_c}\mathbf{y}_k,$$

$$\mathbf{y}_{k+1} = \mathbf{Z}_3^{n_c}\mathbf{y}_k + \mathbf{Z}_4^{n_c}(\nabla\mathbf{f}(\mathbf{x}_{k+1}) - \nabla\mathbf{f}(\mathbf{x}_k))$$

GTA Framework - Multiple Communications

$$n_c \rightarrow \#\text{of communication steps}$$

$$\mathbf{W}_i \rightarrow \mathbf{W}_i^{n_c} \qquad \forall i = 1,2,3,4$$

$$\beta_i \rightarrow \beta_i^{n_c} \qquad \forall i = 1,2,3,4$$

$$\mathbf{Z}_i \rightarrow \mathbf{Z}_i^{n_c} = \mathbf{W}_i^{n_c} \otimes I_d \qquad \forall i = 1,2,3,4$$

$$\mathbf{x}_{k+1} = \mathbf{Z}_1^{n_c}\mathbf{x}_k - \alpha\mathbf{Z}_2^{n_c}\mathbf{y}_k,$$

$$\mathbf{y}_{k+1} = \mathbf{Z}_3^{n_c}\mathbf{y}_k + \mathbf{Z}_4^{n_c}(\nabla\mathbf{f}(\mathbf{x}_{k+1}) - \nabla\mathbf{f}(\mathbf{x}_k))$$

With more communcation, i.e., increase in n_c

- The step size condition increases
- The rate of convergence decreases

GTA Multiple Communications - Rate of Convergence

Theorem

Suppose Assumption 1 holds, number of communications is at least 1 ($n_c \geq 1$) and $\alpha \leq \frac{1}{L}$, iterates $\{x_k, y_k\}$ converge to the solution at a linear rate upper bounded by the following expressions

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

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

For GTA-3 $\max\left\{1 - \frac{\alpha\mu}{2},\ \beta^{n_c}\left(1 + \sqrt{\alpha L}\left(2.5 + \sqrt{\kappa}\right)\right)\right\}$

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

GTA Multiple Communications - Numerical Experiments

Cyclic Network $\beta = 0.992$

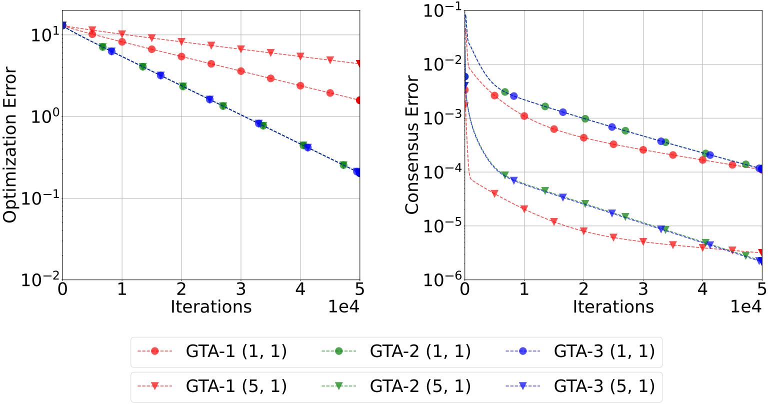


Figure: Quadratics, $n = 16$, $d = 10$, $\kappa = 10^4$

Multiple Communications and Computations

- **GTA** (Gradient Tracking Algorithmic Framework)
  Deterministic
  $n_g \rightarrow \#$ of computation steps
  $n_c \rightarrow \#$ of communication steps
  Berahas, Bollapragada and Gupta (2023). *Balancing Communication and Computation in Gradient Tracking Algorithms for Decentralized Optimization*

- **RGTA** (Randomized Gradient Tracking Framework)
  Randomized
  1 computation step
  n_c communication steps $\rightarrow$ with probability p
  Berahas, Bollapragada and Gupta (2023). *A Flexible Gradient Tracking Algorithmic Framework for Decentralized Optimization* (coming very soon)

GTA - Multiple Communications and Computations

$n_c \rightarrow$ # of communication steps
$n_g \rightarrow$ # of computation steps

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

GTA - Multiple Communications and Computations

$n_c \rightarrow$ # of communication steps
$n_g \rightarrow$ # of computation steps

$$\mathbf{x}_{k+1,1} = \mathbf{Z}_1^{n_c}\mathbf{x}_{k,n_g} - \alpha\mathbf{Z}_2^{n_c}\mathbf{y}_{k,n_g}$$

$$\mathbf{y}_{k+1,1} = \mathbf{Z}_3^{n_c}\mathbf{y}_{k,n_g} + \mathbf{Z}_4^{n_c}(\nabla\mathbf{f}(\mathbf{x}_{k+1,1}) - \nabla\mathbf{f}(\mathbf{x}_{k,n_g}))$$

For $j \rightarrow 1, 2, ..., n_g - 1$

$$\mathbf{x}_{k+1,j+1} = \mathbf{x}_{k+1,j} - \alpha\mathbf{y}_{k+1,j},$$

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

GTA - Multiple Communications and Computations

$n_c \rightarrow \#$ of communication steps
$n_g \rightarrow \#$ of computation steps

$$\mathbf{x}_{k+1,1} = \mathbf{Z}_1^{n_c}\mathbf{x}_{k,n_g} - \alpha\mathbf{Z}_2^{n_c}\mathbf{y}_{k,n_g}$$

$$\mathbf{y}_{k+1,1} = \mathbf{Z}_3^{n_c}\mathbf{y}_{k,n_g} + \mathbf{Z}_4^{n_c}(\nabla\mathbf{f}(\mathbf{x}_{k+1,1}) - \nabla\mathbf{f}(\mathbf{x}_{k,n_g}))$$

For $j \rightarrow 1, 2, ..., n_g - 1$

$$\mathbf{x}_{k+1,j+1} = \mathbf{x}_{k+1,j} - \alpha\mathbf{y}_{k+1,j},$$

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

Theorem

Under previous assumptions, $\beta_1, \beta_3 < 1$, number of communication steps is at least one ($n_c \geq 1$) and number of computation steps is finite ($1 \leq n_g < \infty$), then $\exists\ \alpha > 0$, s.t. the iterates $\{x_k, y_k\}$ converge to the solution at a linear rate.

GTA Multiple Communications and Computations - Numerical Experiments

Cyclic Network $\beta = 0.992$

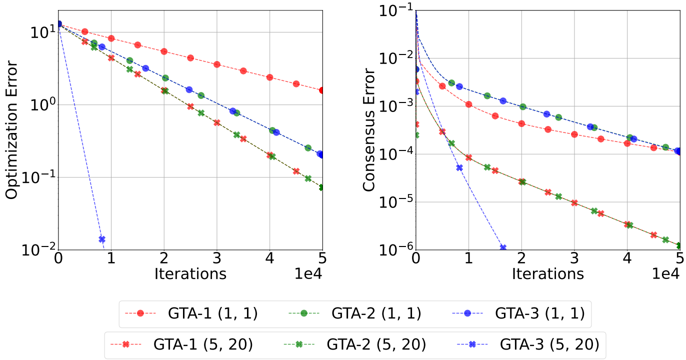


Figure: Quadratics, $n = 16$, $d = 10$, $\kappa = 10^4$

RGTA - Randomized Gradient Tracking Algorithm

Performing communications less often randomly

With probability p:

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

Else:

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

RGTA - Randomized Gradient Tracking Algorithm

Performing communications less often randomly

With probability p:

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

Else:

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

Theorem

Under previous assumptions, $\beta_1, \beta_3 < 1$, number of communication steps is at least one ($n_c \geq 1$) and probability of communication ($0 < p \leq 1$), then $\exists\ \alpha > 0$, s.t. the iterates $\{x_k, y_k\}$ converge to the solution at a linear rate in expectation.

RGTA

- Computation Complexity
  - Decreases as p increases
  - Decreases as n_c increases and then platues

- Communication Complexity
  - $\exists \quad 0 < p^* < 1$ that minimizes the communication complexity
  - $\exists \quad n_c^* \geq 1$ that minimizes the communication complexity

RGTA Multiple Communications and Computations - Numerical Experiments

Star Network $\beta = 0.95$

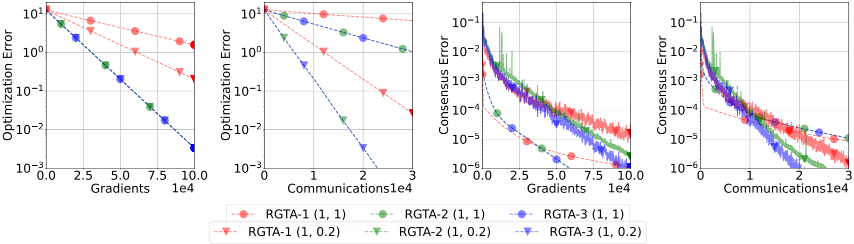


Figure: Quadratics, $n = 16$, $d = 10$, $\kappa = 10^4$

Conclusions

1. We provide a unifying gradient tracking algorithmic framework that allows performing theoretical comparisons between different gradient tracking methods.

2. We provide the flexibility to perform any composition of communication and computation steps in each iteration and show linear rate of convergence.

3. Adapting your algorithm to the system with this flexibility can allow you to improve overall efficiency.

Thank You!

Questions?