# Balancing Computation and Communication in Distributed Optimization

Ermin Wei

Department of Electrical Engineering and Computer Science
Northwestern University

DIMACS
Rutgers Univeristy

Aug 23, 2017

# Collaborators

Albert S. Berahas

Raghu Bollapragada

Nitish Shirish Keskar

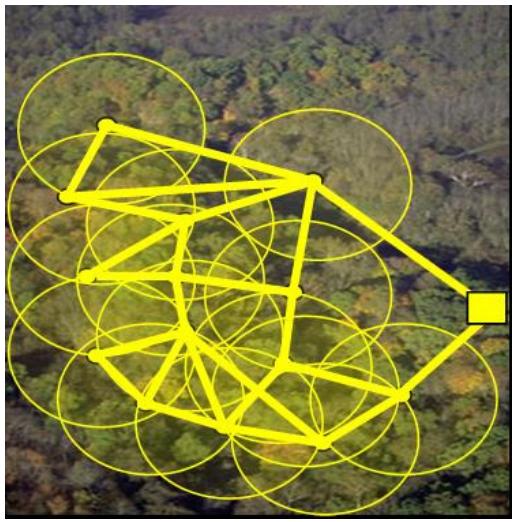
# Overview

# Overview

# Problem Formulation

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

- Applications: Sensor Networks, Robotic Teams, Machine Learning.



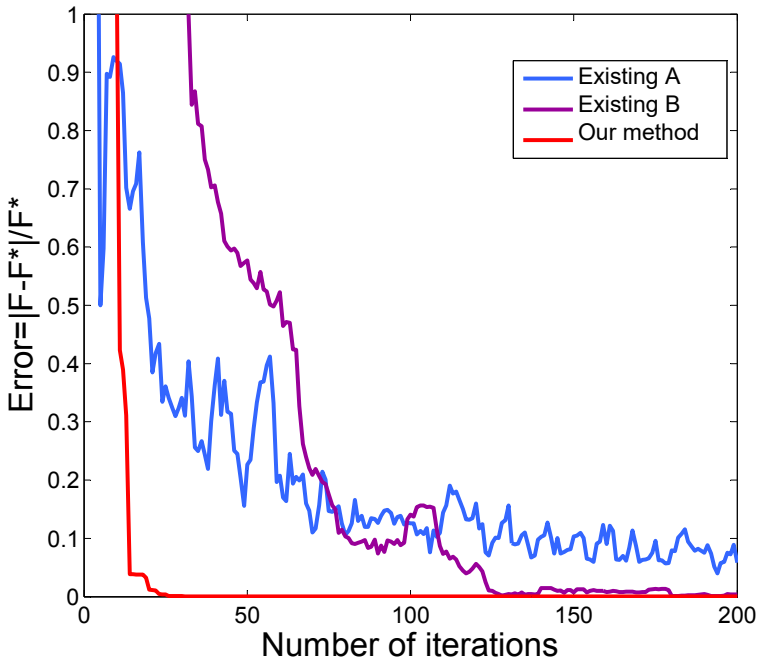
Parameter estimation in sensor networks.
Communication

Multi-agent cooperative control and coordination.
Battery

Large scale computation.
Computation

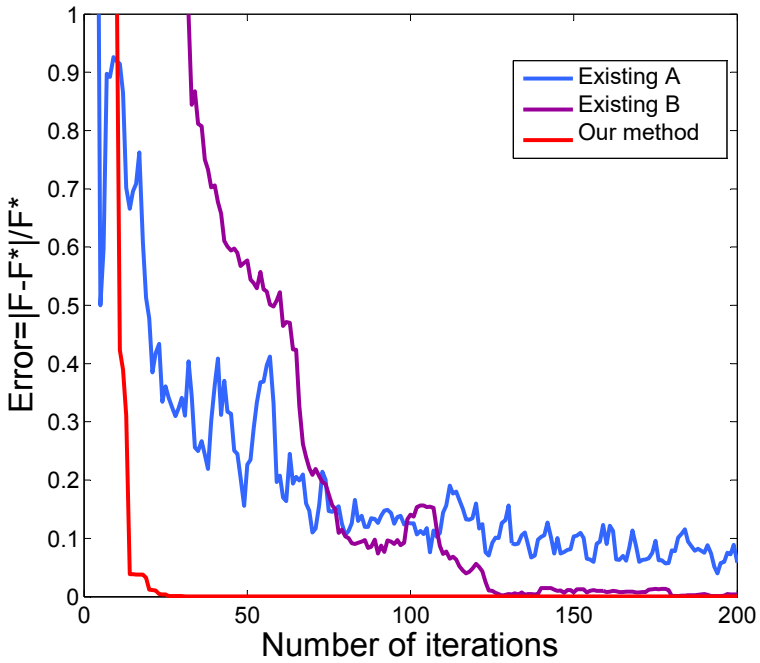
# Algorithm Evaluation

- Typical numerical results (measured in iterations or time or communication rounds).



# Algorithm Evaluation

- Typical numerical results (measured in iterations or time or communication rounds).




- Evaluation framework should reflect features of different applications.

# Problem Formulation

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

- Applications: Sensor Networks, Robotic Teams, Machine Learning.

# Problem Formulation

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

- Applications: Sensor Networks, Robotic Teams, Machine Learning.
- Distributed Setting:

# Problem Formulation

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

- Applications: Sensor Networks, Robotic Teams, Machine Learning.
- Distributed Setting: Consensus Optimization problem

$$\min_{x_i \in \mathbb{R}^p} f(x) = \sum_{i=1}^{n} f_i(x_i)$$
$$\text{s.t.} \quad x_i = x_j, \quad \forall i, j \in \mathcal{N}_i$$

- each node $i$ has a local copy of the parameter vector $x_i$
- @ optimality consensus is achieved among all the nodes in the network

# Consensus Optimization Problem

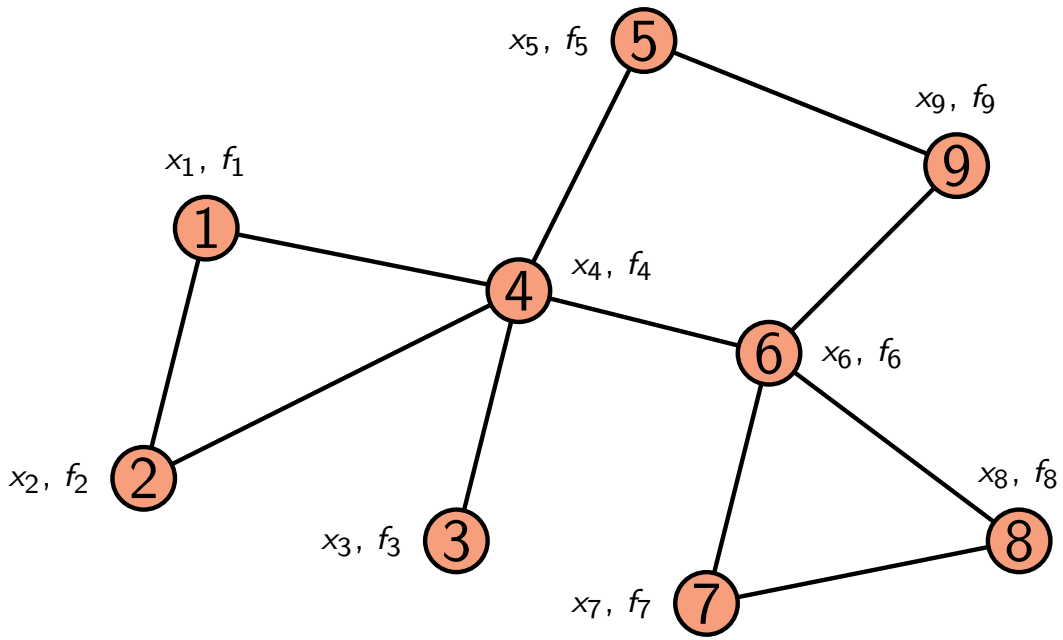
$$\min_{x_i \in \mathbb{R}^p} f(x) = \sum_{i=1}^{n} f_i(x_i)$$

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

# Consensus Optimization Problem

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

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



# Consensus Optimization Problem

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

$$\text{s.t.} \quad \mathbf{Zx} = \mathbf{x}$$

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

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

# Literature Review

1. Sublinearly Converging Methods
   - DGD [Tsitsiklis and Bertsekas, 1989; Nedić and Ozdaglar, 2009; Sundhar Ram et. al., 2010; Tsianos and Rabbat, 2012; Yuan et. al., 2015; Zeng and Yin, 2016] ...
2. Linearly and Superlinearly Converging Methods
   - EXTRA [Shi et. al., 2015], DIGing [Nedić et. al., 2017], NEXT [Lorenzo and Scutari, 2015], Aug-DGM [Xu et. al., 2015], NN-EXTRA [Mokhtari et. al., 2016], [Qu and Li, 2017], DQN [Eisen et. al., 2017], NN [Mokhtari et. al., 2014, 2015]...
3. Communication Efficient Methods
   - [Chen and Ozdaglar, 2012], [Shamir et. al., 2014], [Chow et. al., 2016], [Lan et. al., 2017], [Tsianos et. al., 2012], [Zhang and Lin, 2015], ...
4. Asynchronous Methods
   - [Tsitsiklis and Bertsekas, 1989], [Tsitsiklis et. al., 1986], [Sundhar Ram et. al., 2009], [Wei and Ozdaglar, 2013], [Mansoori and Wei, 2017], [Zhang and Kwok, 2014], [Wu et. al., 2017], ...

# Literature Review

1. Sublinearly Converging Methods
   - DGD [Tsitsiklis and Bertsekas, 1989; Nedić and Ozdaglar, 2009; Sundhar Ram et. al., 2010; Tsianos and Rabbat, 2012; Yuan et. al., 2015; Zeng and Yin, 2016] ...
2. Linearly and Superlinearly Converging Methods
   - EXTRA [Sh NEXT [Lorenzo and Scutari, 201 [Mokhtari et. al., 2016], [Qu and Li, ari et. al., 2014, 2015]...
3. Communicatio
   - [Chen and v et. al., 2016], [Lan et. al., 2017], [Tsianos et. al., 2012], [Zhang and Lin, 2015], ...
4. Asynchronous Methods
   - [Tsitsiklis and Bertsekas, 1989], [Tsitsiklis et. al., 1986], [Sundhar Ram et. al., 2009], [Wei and Ozdaglar, 2013], [Mansoori and Wei, 2017], [Zhang and Kwok, 2014], [Wu et. al., 2017], ...

Communication
Computation

# Literature Review

1. Sublinearly Converging Methods
   - DGD [Tsitsiklis and Bertsekas, 1989; Nedić and Ozdaglar, 2009; Sundhar Ram et. al., 2010; Tsianos and Rabbat, 2012; Yuan et. al., 2015; Zeng and Yin, 2016] ...
2. Linearly and Superlinearly Converging Methods
   - EXTRA [Sh... NEXT [Lorenzo and Scutari, 201... [Mokhtari et. al., 2016], [Qu and Li,... ari et. al., 2014, 2015]...
3. Communicatio...
   - [Chen and ... et. al., 2016], [Lan et. al., 2017], [Tsianos et. al., 2012], [Zhang and Lin, 2015], ...
4. Asynchronous Methods
   - [Tsitsiklis and Bertsekas, 1989], [Tsitsiklis et. al., 1986], [Sundhar Ram et. al., 2009], [Wei and Ozdaglar, 2013], [Mansoori and Wei, 2017], [Zhang and Kwok, 2014], [Wu et. al., 2017], ...

$$\text{Cost} = \#\text{Communications} + \#\text{Computations}$$

# Literature Review

1. Sublinearly Converging Methods
   - DGD [Tsitsiklis and Bertsekas, 1989; Nedić and Ozdaglar, 2009; Sundhar Ram et. al., 2010; Tsianos and Rabbat, 2012; Yuan et. al., 2015; Zeng and Yin, 2016] ...
2. Linearly and Superlinearly Converging Methods
   - EXTRA [Sh NEXT [Lorenzo and Scutari, 201 [Mokhtari et. al., 2016], [Qu and Li, ari et. al., 2014, 2015]...
3. Communicatio
   - [Chen and v et. al., 2016], [Lan et. al., 2017], [Tsianos et. al., 2012], [Zhang and Lin, 2015], ...
4. Asynchronous Methods
   - [Tsitsiklis and Bertsekas, 1989], [Tsitsiklis et. al., 1986], [Sundhar Ram et. al., 2009], [Wei and Ozdaglar, 2013], [Mansoori and Wei, 2017], [Zhang and Kwok, 2014], [Wu et. al., 2017], ...

$$\text{Cost} = \#\text{Communications} \times c_c + \#\text{Computations} \times c_g$$

# Goal of the Project

- Develop an algorithmic framework that is independent of the method used to balance computation and communication in distributed optimization

# Goal of the Project

- Develop an algorithmic framework that is independent of the method used to balance computation and communication in distributed optimization
- Prove convergence for methods that use the framework

# Goal of the Project

- Develop an algorithmic framework that is independent of the method used to balance computation and communication in distributed optimization
- Prove convergence for methods that use the framework
- Show that the framework can be applied to a many consensus optimization problems (with different communication and computation costs)

# Goal of the Project

- Develop an algorithmic framework that is independent of the method used to balance computation and communication in distributed optimization
- Prove convergence for methods that use the framework
- Show that the framework can be applied to a many consensus optimization problems (with different communication and computation costs)
- Illustrate empirically that methods that utilize the framework outperform their base algorithms for specific applications

# This talk

- First stage of the project
- Multiple consensus in DGD (theoretically and in practice)
- Design a flexible first-order algorithm that decouples the two operations
- Investigate the method theoretically and empirically
- By-product: variants of DGD with exact convergence

# This talk

- First stage of the project
- Multiple consensus in DGD (theoretically and in practice)
- Design a flexible first-order algorithm that decouples the two operations
- Investigate the method theoretically and empirically
- By-product: variants of DGD with exact convergence

Not in this talk (ongoing work)

- Multiple gradients
- Extend framework to different algorithms (e.g., EXTRA, NN) or asynchronous methods

# Overview

# Distributed Gradient Descent (DGD)

**DGD** [Tsitsiklis and Bertsekas, 1989; Nedić and Ozdaglar, 2009; Sundhar Ram et. al., 2010; Tsianos and Rabbat, 2012; Yuan et. al., 2015; Zeng and Yin, 2016]

$$x_{i,k+1} = \sum_{j \in \mathcal{N}_i \cup i} w_{ij} x_{j,k} - \alpha \nabla f_i(x_{i,k}), \qquad \forall i = 1, ..., n$$

# Distributed Gradient Descent (DGD)

**DGD** [Tsitsiklis and Bertsekas, 1989; Nedić and Ozdaglar, 2009; Sundhar Ram et. al., 2010; Tsianos and Rabbat, 2012; Yuan et. al., 2015; Zeng and Yin, 2016]

$$x_{i,k+1} = \sum_{j \in \mathcal{N}_i \cup i} w_{ij} x_{j,k} - \alpha \nabla f_i(x_{i,k}), \qquad \forall i = 1, ..., n$$

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

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} \in \mathbb{R}^{np}, \quad \nabla \mathbf{f}(\mathbf{x}_k) = \begin{bmatrix} \nabla f_1(x_{1,k}) \\ \nabla f_2(x_{2,k}) \\ \vdots \\ \nabla f_n(x_{n,k}) \end{bmatrix} \in \mathbb{R}^{np}, \quad \mathbf{Z} = \mathbf{W} \otimes I_p \in \mathbb{R}^{np \times np}$$

# Distributed Gradient Descent (DGD)

**DGD** [Tsitsiklis and Bertsekas, 1989; Nedić and Ozdaglar, 2009; Sundhar Ram et. al., 2010; Tsianos and Rabbat, 2012; Yuan et. al., 2015; Zeng and Yin, 2016]

$$x_{i,k+1} = \sum_{j \in \mathcal{N}_i \cup i} w_{ij} x_{j,k} - \alpha \nabla f_i(x_{i,k}), \qquad \forall i = 1, ..., n$$

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

- Diminishing $\alpha$: Sub-linear convergence to the solution
- Constant $\alpha$: Linear convergence to a neighborhood $\mathcal{O}(\alpha)$

# DGD – Questions

**DGD**

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

# DGD – Questions

**DGD**

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

- Is it necessary to do one consensus step and one optimization (gradient) step?

# DGD – Questions

## DGD

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

- Is it necessary to do one consensus step and one optimization (gradient) step?
- If not, what is the interpretation of methods that do more consensus/optimization steps?
- What convergence guarantees can be proven for such methods?
- How do these variants perform in practice?

# DGD – Questions

## DGD

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

- Is it necessary to do one consensus step and one optimization (gradient) step?
- If not, what is the interpretation of methods that do more consensus/optimization steps?
- What convergence guarantees can be proven for such methods?
- How do these variants perform in practice?

## DGD$^t$

$$\mathbf{x}_{k+1} = \mathbf{Z}^t\mathbf{x}_k - \alpha\nabla\mathbf{f}(\mathbf{x}_k), \qquad \mathbf{Z}^t = \mathbf{W}^t \otimes I_p$$

# DGD – Assumptions & Definitions

## Assumptions

1. Each component function $f_i$ is $\mu_i > 0$ strongly convex and has $L_i > 0$ Lipschitz continuous gradients
2. The mixing matrix $\mathbf{W}$ is symmetric and doubly stochastic with $\beta < 1$ ($\beta$ is the second largest eigenvalue)

## Definitions

$$\bar{x}_k = \frac{1}{n}\sum_{i=1}^{n} x_{i,k}, \quad \nabla f(x_k) = \sum_{i=1}^{n} \nabla f_i(x_{i,k}), \quad \nabla f(\bar{x}_k) = \sum_{i=1}^{n} \nabla f_i(\bar{x}_k)$$

# DGD – Theory – Bounded distance to minimum

## Theorem (Bounded distance to minimum) [Yuan et. al., 2015]

Suppose Assumptions 1 & 2 hold, and let the step length satisfy

$$\alpha \leq \min\left\{\frac{1+\lambda_n(\mathbf{W})}{L_f}, \frac{1}{\mu_f + L_f}\right\}$$

where $\mu_f$ is the strong convexity parameter of $f$ and $L_f$ is the Lipschitz constant of the gradient of $f$. Then, for all $k = 0, 1, \ldots$

$$\|\bar{x}_{k+1} - x^\star\|^2 \leq c_1^2\|\bar{x}_k - x^\star\|^2 + \frac{c_3^2}{(1-\beta)^2}$$

$$c_1^2 = 1 - \alpha c_2 + \alpha\delta - \alpha^2\delta c_2, \quad c_2 = \frac{\mu_f L_f}{\mu_f + L_f},$$

$$c_3^2 = \alpha^3(\alpha + \delta^{-1})L^2D^2, \quad D = \sqrt{2L(\sum_{i=1}^{n} f_i(0) - f^\star)},$$

where $x^\star = \arg\min_x f(x)$ and $\delta > 0$.

# DGD – Theory – Bounded distance to minimum

## Theorem (Bounded distance to minimum) [Yuan et. al., 2015]

Suppose Assumptions 1 & 2 hold, and let the step length satisfy

$$\alpha \leq \min\left\{\frac{1+\lambda_n(\mathbf{W})}{L_f}, \frac{1}{\mu_f + L_f}\right\}$$

where $\mu_f$ is the strong convexity parameter of $f$ and $L_f$ is the Lipschitz constant of the gradient of $f$. Then, for all $k = 0, 1, \ldots$

$$\|\bar{x}_{k+1} - x^\star\|^2 \leq c_1^2\|\bar{x}_k - x^\star\|^2 + \frac{\boldsymbol{c_3^2}}{\boldsymbol{(1-\beta)^2}}$$

$$c_1^2 = 1 - \alpha c_2 + \alpha\delta - \alpha^2\delta c_2, \quad c_2 = \frac{\mu_f L_f}{\mu_f + L_f},$$

$$c_3^2 = \alpha^3(\alpha + \delta^{-1})L^2D^2, \quad D = \sqrt{2L(\sum_{i=1}^{n} f_i(0) - f^\star)},$$

where $x^\star = \arg\min_x f(x)$ and $\delta > 0$.

# DGD$^t$ – Theory – Bounded distance to minimum

## Theorem (Bounded distance to minimum) [ASB, RB, NSK and EW, 2017]

Suppose Assumptions 1 & 2 hold, and let the step length satisfy

$$\alpha \leq \min\left\{\frac{1+\lambda_n(\mathbf{W}^t)}{L_f}, \frac{1}{\mu_f + L_f}\right\}$$

where $\mu_f$ is the strong convexity parameter of $f$ and $L_f$ is the Lipschitz constant of the gradient of $f$. Then, for all $k = 0, 1, \ldots$

$$\|\bar{x}_{k+1} - x^\star\|^2 \leq c_1^2\|\bar{x}_k - x^\star\|^2 + \frac{c_3^2}{(1-\beta^t)^2}$$

$$c_1^2 = 1 - \alpha c_2 + \alpha\delta - \alpha^2\delta c_2, \quad c_2 = \frac{\mu_f L_f}{\mu_f + L_f},$$

$$c_3^2 = \alpha^3(\alpha + \delta^{-1})L^2D^2, \quad D = \sqrt{2L(\sum_{i=1}^n f_i(0) - f^\star)},$$

where $x^\star = \arg\min_x f(x)$ and $\delta > 0$.

# DGD – Theory – Comments

- Can show similar error neighborhood for $\|x_{i,k} - x^*\|$.
- Theoretical results are similar to DGD in nature (constant $\alpha$)
  - Linear convergence to a neighborhood of the solution
  - Improved neighborhood

$$\mathcal{O}\left(\frac{1}{(1-\beta)^2}\right) \quad \text{v.s.} \quad \mathcal{O}\left(\frac{1}{(1-\beta^t)^2}\right)$$

  - But, cannot kill the neighborhood with increased communication

# DGD – Theory – Comments

- Can show similar error neighborhood for $\|x_{i,k} - x^*\|$.
- Theoretical results are similar to DGD in nature (constant $\alpha$)
  - Linear convergence to a neighborhood of the solution
  - Improved neighborhood

$$\mathcal{O}\left(\frac{1}{(1-\beta)^2}\right) \quad \text{v.s.} \quad \mathcal{O}\left(\frac{1}{(1-\beta^t)^2}\right)$$

  - But, cannot kill the neighborhood with increased communication
- Drawback: requires extra communication

# DGD – Theory – Comments

- Can show similar error neighborhood for $\|x_{i,k} - x^*\|$.
- Theoretical results are similar to DGD in nature (constant $\alpha$)
  - Linear convergence to a neighborhood of the solution
  - Improved neighborhood

$$\mathcal{O}\left(\frac{1}{(1-\beta)^2}\right) \quad \text{v.s.} \quad \mathcal{O}\left(\frac{1}{(1-\beta^t)^2}\right)$$

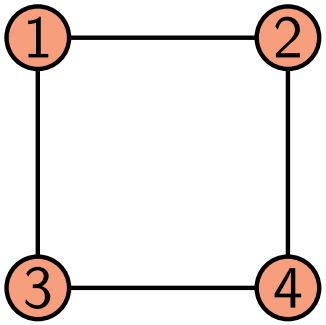
  - But, cannot kill the neighborhood with increased communication
- Drawback: requires extra communication
- Effectively, DGD$^t$ is DGD with a different underlying graph (different weights in $\mathbf{W}$)

# DGD – Theory – Comments

- Effectively, DGD$^t$ is DGD with a different underlying graph (different weights in $\mathbf{W}$)

# DGD – Theory – Comments

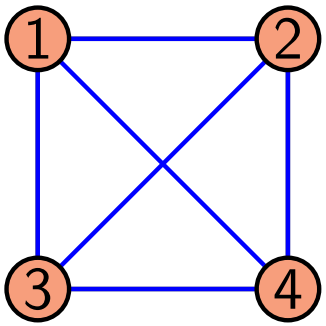
- Effectively, DGD$^t$ is DGD with a different underlying graph (different weights in $\mathbf{W}$)




$$W = \begin{bmatrix} 1/2 & 1/4 & 1/4 & 0 \\ 1/4 & 1/2 & 0 & 1/4 \\ 1/4 & 0 & 1/2 & 1/4 \\ 0 & 1/4 & 1/4 & 1/2 \end{bmatrix},$$

# DGD – Theory – Comments

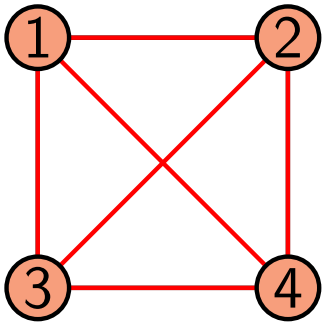
- Effectively, DGD$^t$ is DGD with a different underlying graph (different weights in $\mathbf{W}$)




$$W = \begin{bmatrix} 1/2 & 1/4 & 1/4 & 0 \\ 1/4 & 1/2 & 0 & 1/4 \\ 1/4 & 0 & 1/2 & 1/4 \\ 0 & 1/4 & 1/4 & 1/2 \end{bmatrix}, \quad W^2 = \begin{bmatrix} 3/8 & 1/4 & 1/4 & \frac{1}{8} \\ 1/4 & 3/8 & 1/8 & 1/4 \\ 1/4 & 1/8 & 3/8 & 1/4 \\ 1/8 & 1/4 & 1/4 & 3/8 \end{bmatrix}$$

# DGD – Theory – Comments

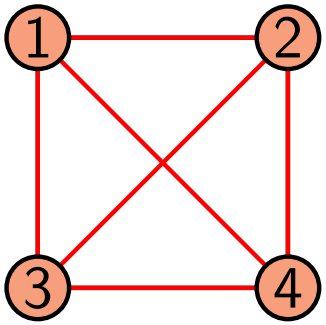
- Effectively, DGD$^t$ is DGD with a different underlying graph (different weights in $\mathbf{W}$)



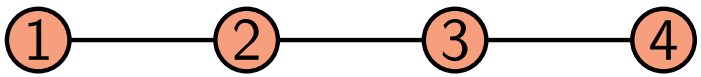

$$W = \begin{bmatrix} 1/2 & 1/4 & 1/4 & 0 \\ 1/4 & 1/2 & 0 & 1/4 \\ 1/4 & 0 & 1/2 & 1/4 \\ 0 & 1/4 & 1/4 & 1/2 \end{bmatrix}, \quad W^{10} \approx \begin{bmatrix} 1/4 & 1/4 & 1/4 & 1/4 \\ 1/4 & 1/4 & 1/4 & 1/4 \\ 1/4 & 1/4 & 1/4 & 1/4 \\ 1/4 & 1/4 & 1/4 & 1/4 \end{bmatrix}$$

# DGD – Theory – Comments

- Effectively, DGD$^t$ is DGD with a different underlying graph (different weights in $\mathbf{W}$)



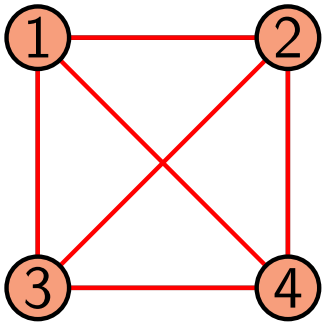

$$W = \begin{bmatrix} 1/2 & 1/4 & 1/4 & 0 \\ 1/4 & 1/2 & 0 & 1/4 \\ 1/4 & 0 & 1/2 & 1/4 \\ 0 & 1/4 & 1/4 & 1/2 \end{bmatrix}, \quad W^{10} \approx \begin{bmatrix} 1/4 & 1/4 & 1/4 & 1/4 \\ 1/4 & 1/4 & 1/4 & 1/4 \\ 1/4 & 1/4 & 1/4 & 1/4 \\ 1/4 & 1/4 & 1/4 & 1/4 \end{bmatrix}$$



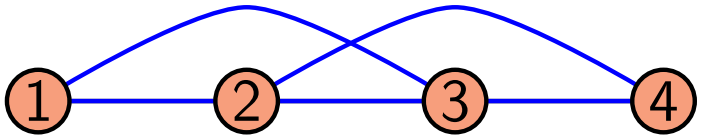

$$W = \begin{bmatrix} 2/3 & 1/3 & 0 & 0 \\ 1/3 & 1/3 & 1/3 & 0 \\ 0 & 1/3 & 1/3 & 1/3 \\ 0 & 0 & 1/3 & 2/3 \end{bmatrix},$$

# DGD – Theory – Comments

- Effectively, DGD$^t$ is DGD with a different underlying graph (different weights in $\mathbf{W}$)



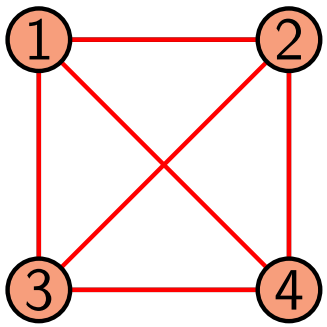

$$W = \begin{bmatrix} 1/2 & 1/4 & 1/4 & 0 \\ 1/4 & 1/2 & 0 & 1/4 \\ 1/4 & 0 & 1/2 & 1/4 \\ 0 & 1/4 & 1/4 & 1/2 \end{bmatrix}, \quad W^{10} \approx \begin{bmatrix} 1/4 & 1/4 & 1/4 & 1/4 \\ 1/4 & 1/4 & 1/4 & 1/4 \\ 1/4 & 1/4 & 1/4 & 1/4 \\ 1/4 & 1/4 & 1/4 & 1/4 \end{bmatrix}$$



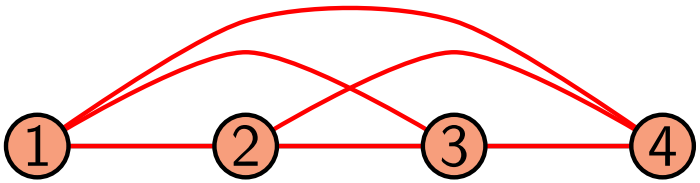

$$W = \begin{bmatrix} 2/3 & 1/3 & 0 & 0 \\ 1/3 & 1/3 & 1/3 & 0 \\ 0 & 1/3 & 1/3 & 1/3 \\ 0 & 0 & 1/3 & 2/3 \end{bmatrix}, \quad W^2 = \begin{bmatrix} 5/9 & 1/3 & 1/9 & 0 \\ 1/3 & 1/3 & 2/9 & 1/9 \\ 1/9 & 2/9 & 1/3 & 1/3 \\ 0 & 1/9 & 1/3 & 5/9 \end{bmatrix}$$

# DGD – Theory – Comments

- Effectively, DGD$^t$ is DGD with a different underlying graph (different weights in $\mathbf{W}$)




$$W = \begin{bmatrix} 1/2 & 1/4 & 1/4 & 0 \\ 1/4 & 1/2 & 0 & 1/4 \\ 1/4 & 0 & 1/2 & 1/4 \\ 0 & 1/4 & 1/4 & 1/2 \end{bmatrix}, \quad W^{10} \approx \begin{bmatrix} 1/4 & 1/4 & 1/4 & 1/4 \\ 1/4 & 1/4 & 1/4 & 1/4 \\ 1/4 & 1/4 & 1/4 & 1/4 \\ 1/4 & 1/4 & 1/4 & 1/4 \end{bmatrix}$$




$$W = \begin{bmatrix} 2/3 & 1/3 & 0 & 0 \\ 1/3 & 1/3 & 1/3 & 0 \\ 0 & 1/3 & 1/3 & 1/3 \\ 0 & 0 & 1/3 & 2/3 \end{bmatrix}, \quad W^3 = \begin{bmatrix} 0.48 & 0.33 & 0.15 & 0.04 \\ 0.33 & 0.30 & 0.22 & 0.15 \\ 0.15 & 0.22 & 0.30 & 0.33 \\ 0.04 & 0.15 & 0.33 & 0.48 \end{bmatrix}$$

# DGD$^{(t)}$ – Numerical Results

- **Problem**: Quadratic

$$f(x) = \frac{1}{2}\sum_{i=1}^{n} x^T A_i x + b_i^T x$$

  each node $i = \{1, ..., n\}$ has local data $A_i \in \mathbb{R}^{n_i \times p}$ and $b_i \in \mathbb{R}^{n_i}$
- **Parameters**: $n = 10$, $p = 10$, $n_i = 10$, $\kappa = 10^2$
- **Methods**: DGD (1,1), DGD (1,2), DGD (1,5), DGD (1,10)
- **Graph**: 4-cyclic graph, $w_{ii} = \frac{1}{5}$, $w_{ij} = \begin{cases} \frac{1}{5} & \text{if } j \in \mathcal{N}_i \\ 0 & \text{otherwise} \end{cases}$

# DGD$^{(t)}$ – Numerical Results

- **Problem**: Quadratic

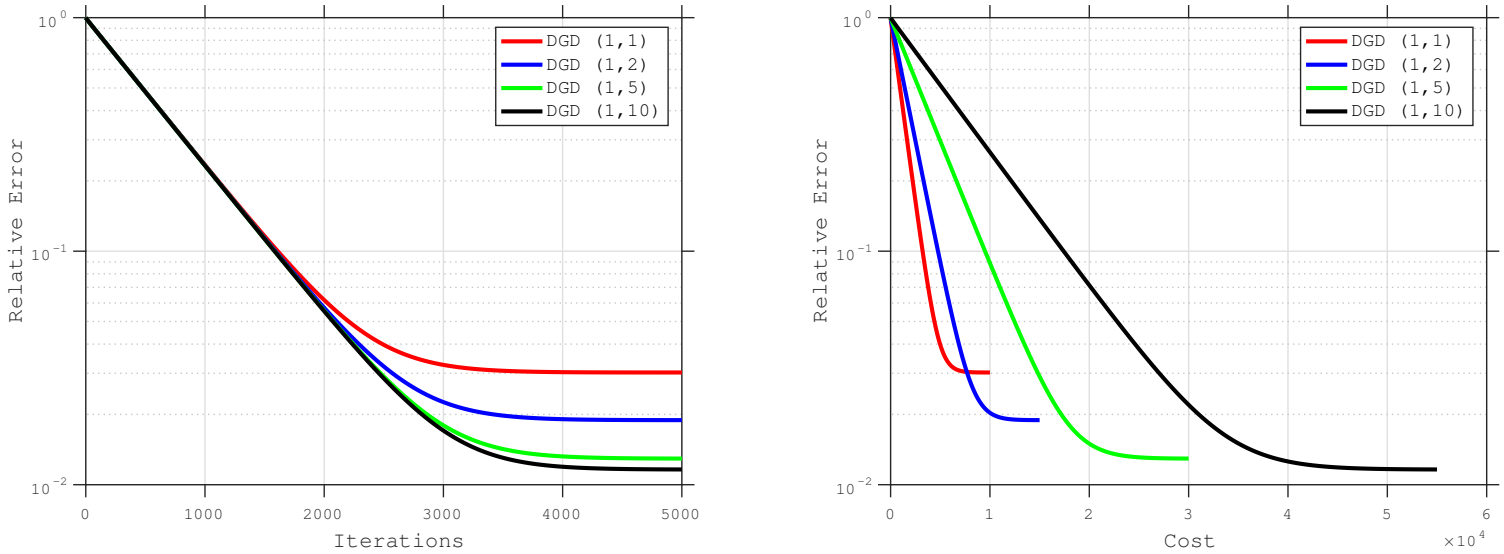
$$f(x) = \frac{1}{2}\sum_{i=1}^{n} x^T A_i x + b_i^T x$$

  each node $i = \{1, ..., n\}$ has local data $A_i \in \mathbb{R}^{n_i \times p}$ and $b_i \in \mathbb{R}^{n_i}$

- **Parameters**: $n = 10$, $p = 10$, $n_i = 10$, $\kappa = 10^2$
- **Methods**: DGD (1,1), DGD (1,2), DGD (1,5), DGD (1,10)
- **Graph**: 4-cyclic graph, $w_{ii} = \frac{1}{5}$, $w_{ij} = \begin{cases} \frac{1}{5} & \text{if } j \in \mathcal{N}_i \\ 0 & \text{otherwise} \end{cases}$

**Show the effect of multiple consensus steps per gradient step**

# DGD$^{(t)}$ – Numerical Results




Quadratic. $n = 10$, $p = 10$, $n_i = 10$, $\kappa = 10^2$.

$Cost = \#Communications \times 1 + \#Computations \times 1$

# Overview

# Operators

## DGD

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

# Operators

## DGD

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

### Operators

- $\mathcal{W}[\mathbf{x}] = \mathbf{Z}\mathbf{x}$
- $\mathcal{T}[\mathbf{x}] = \mathbf{x} - \alpha\nabla\mathbf{f}(\mathbf{x})$

# Operators

## DGD

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

### Operators

- $\mathcal{W}[\mathbf{x}] = \mathbf{Z}\mathbf{x}$
- $\mathcal{T}[\mathbf{x}] = \mathbf{x} - \alpha\nabla\mathbf{f}(\mathbf{x})$

### Methods

- DGD: $(\mathcal{T} - I + \mathcal{W})[\mathbf{x}_k] = \mathbf{Z}\mathbf{x}_k - \alpha\nabla\mathbf{f}(\mathbf{x}_k)$

# Operators

## DGD

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

### Operators

- $\mathcal{W}[\mathbf{x}] = \mathbf{Z}\mathbf{x}$
- $\mathcal{T}[\mathbf{x}] = \mathbf{x} - \alpha\nabla\mathbf{f}(\mathbf{x})$

### Methods

- DGD: $(\mathcal{T} - I + \mathcal{W})[\mathbf{x}_k] = \mathbf{Z}\mathbf{x}_k - \alpha\nabla\mathbf{f}(\mathbf{x}_k)$
- TW: $\mathcal{T}[\mathcal{W}[\mathbf{x}_k]] = \mathbf{Z}\mathbf{x}_k - \alpha\nabla\mathbf{f}(\mathbf{Z}\mathbf{x}_k)$
- WT: $\mathcal{W}[\mathcal{T}[\mathbf{x}_k]] = \mathbf{Z}\mathbf{x}_k - \alpha\mathbf{Z}\nabla\mathbf{f}(\mathbf{x}_k)$

A special case of the algorithm appeared as CTA (Combined then Adapt) and ATC in [Sayed, 13] for quadratic problems

# Operators

## DGD

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

### Operators

- $\mathcal{W}[\mathbf{x}] = \mathbf{Z}\mathbf{x}$
- $\mathcal{T}[\mathbf{x}] = \mathbf{x} - \alpha\nabla\mathbf{f}(\mathbf{x})$

### Methods

- DGD: $(\mathcal{T} - I + \mathcal{W})[\mathbf{x}_k] = \mathbf{Z}\mathbf{x}_k - \alpha\nabla\mathbf{f}(\mathbf{x}_k)$
- TW: $\mathcal{T}[\mathcal{W}[\mathbf{x}_k]] = \mathbf{Z}\mathbf{x}_k - \alpha\nabla\mathbf{f}(\mathbf{Z}\mathbf{x}_k)$
- WT: $\mathcal{W}[\mathcal{T}[\mathbf{x}_k]] = \mathbf{Z}\mathbf{x}_k - \alpha\mathbf{Z}\nabla\mathbf{f}(\mathbf{x}_k)$

# Operators

**TW**

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

Methods

- DGD: $(\mathcal{T} - I + \mathcal{W})[\mathbf{x}_k] = \mathbf{Z}\mathbf{x}_k - \alpha\nabla\mathbf{f}(\mathbf{x}_k)$
- TW: $\mathcal{T}[\mathcal{W}[\mathbf{x}_k]] = \mathbf{Z}\mathbf{x}_k - \alpha\nabla\mathbf{f}(\mathbf{Z}\mathbf{x}_k)$
- WT: $\mathcal{W}[\mathcal{T}[\mathbf{x}_k]] = \mathbf{Z}\mathbf{x}_k - \alpha\mathbf{Z}\nabla\mathbf{f}(\mathbf{x}_k)$

# TW – Methods, Assumptions & Definitions

## Methods

- TW$^{t}$: $t$ (predetermined) consensus steps for every gradient step
- TW$^{+}$: increasing number of consensus steps

## Assumptions

1. Assumptions 1 & 2, same as before

## Definitions

$$\bar{x}_k = \frac{1}{n}\sum_{i=1}^{n} x_{i,k}, \quad \nabla f(y_k) = \sum_{i=1}^{n} \nabla f_i(y_{i,k}), \quad \nabla f(\bar{x}_k) = \sum_{i=1}^{n} \nabla f_i(\bar{x}_k)$$

# $\mathrm{TW}^t$ – Theory – Bounded distance to minimum

## Theorem (Bounded distance to minimum) [ASB, RB, NSK and EW, 2017]

Suppose Assumptions 1 & 2 hold, and let the step length satisfy

$$\alpha \leq \min\left\{\frac{1+\lambda_n(\mathbf{W}^t)}{L_f}, \frac{1}{\mu_f + L_f}\right\}$$

where $\mu_f$ is the strong convexity parameter of $f$ and $L_f$ is the Lipschitz constant of the gradient of $f$. Then, for all $k = 0, 1, \ldots$

$$\|x_{i,k} - x^\star\| \leq c_i^k \|x^\star\| + \frac{c_3}{\sqrt{1-c_1^2}}\beta^t + \beta^t \alpha D$$

where $x^\star = \arg\min_x f(x)$ and $\delta > 0$.

# $TW^+$ Theory (increasing Consensus)

- Can we increase the number of consensus steps and converge to the solution?

# TW$^+$ Theory (increasing Consensus)

- Can we increase the number of consensus steps and converge to the solution?
- Increase $t(k)$ accordingly so that we kill the error term

$$O(\beta^{t(k)})$$

# TW$^+$ Theory (increasing Consensus)

- Can we increase the number of consensus steps and converge to the solution?
- Increase $t(k)$ accordingly so that we kill the error term

$$O(\beta^{t(k)})$$

- Similar idea appeared in [Chen and Ozdaglar, 2012] for nonsmooth problems

# $TW^+$ Theory (increasing Consensus)

- Can we increase the number of consensus steps and converge to the solution?
- Increase $t(k)$ accordingly so that we kill the error term

$$O(\beta^{t(k)})$$

- Similar idea appeared in [Chen and Ozdaglar, 2012] for nonsmooth problems
- Resulting in $TW^{t(k)}$ algorithm with exact convergence: As long as we keep increasing the number of consensus

# TW$^+$ – Theory – Bounded distance to minimum

## Theorem (Bounded distance to minimum)[ASB, RB, NSK and EW, 2017]

Suppose Assumptions 1 & 2 hold, $t(k) = k$ and let the step length satisfy

$$\alpha \leq \min \left\{ \frac{1}{L_f}, \frac{1}{\mu_{\bar{f}} + L_{\bar{f}}} \right\}$$

where $\mu_f$ is the strong convexity parameter of $f$ and $L_f$ is the Lipschitz constant of the gradient of $f$. Then, for all $k = 0, 1, \ldots$

$$\|x_{i,k} - x^\star\| \leq C\rho^k,$$

where $x^\star = \arg\min_x f(x)$ and some constants $C$, $\rho$.

When $t(k) = k$ to reach an $\epsilon-$accurate solution, we need $O(\log(\frac{1}{\epsilon}))$ number of gradient evaluation and $O((\log(\frac{1}{\epsilon}))^2)$ rounds of communication.

# Numerical Experiments

- **Methods**: DGD, TW (1,1,-), TW (10,1,-), TW (1,10,-), TW (1,1,$k$), TW (1,1,500), TW (1,1,1000)

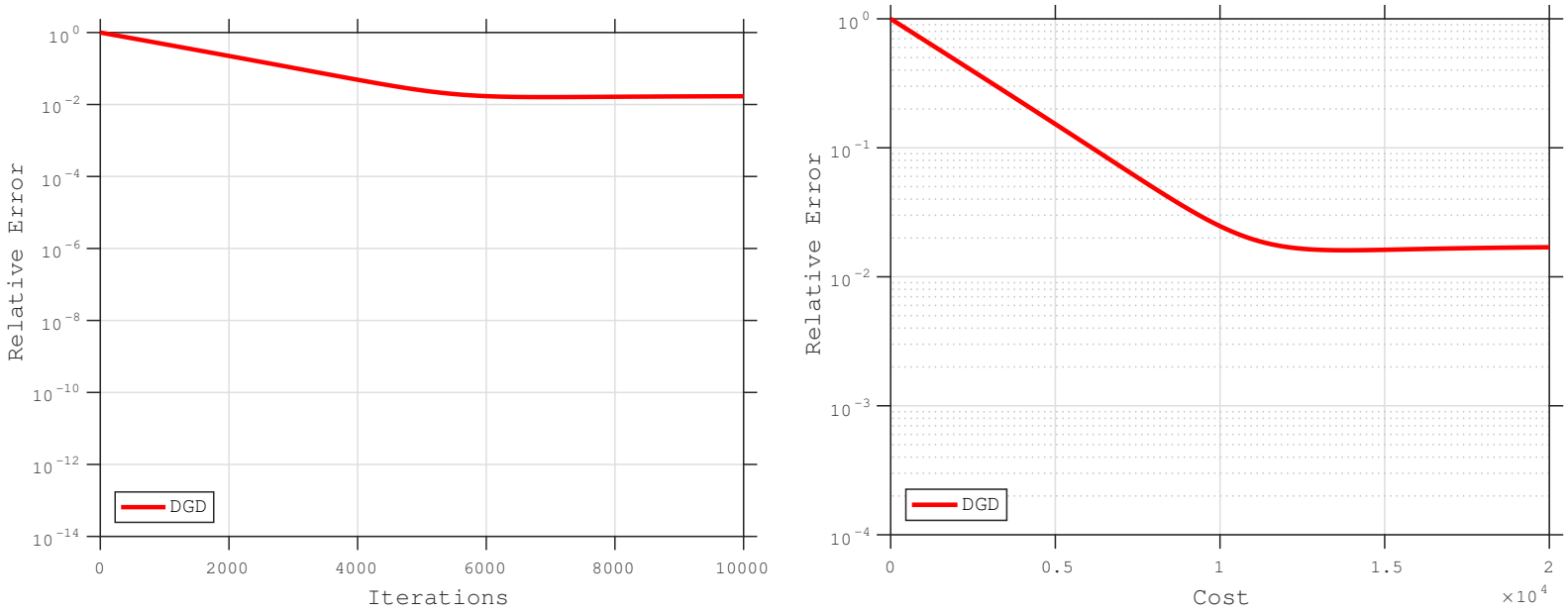
# Numerical Experiments

- **Methods**: DGD, TW (1,1,-), TW (10,1,-), TW (1,10,-), TW (1,1,$k$), TW (1,1,500), TW (1,1,1000)
- **Problem**: Quadratic

$$f(x) = \frac{1}{2}\sum_{i=1}^{n} x^T A_i x + b_i^T x$$

  each node $i = \{1, ..., n\}$ has local data $A_i \in \mathbb{R}^{n_i \times p}$ and $b_i \in \mathbb{R}^{n_i}$
- **Parameters**: $n = 10$, $p = 10$, $n_i = 10$, $\kappa = \frac{L}{\mu} = 10^4$
- **Graph**: 4-cyclic graph

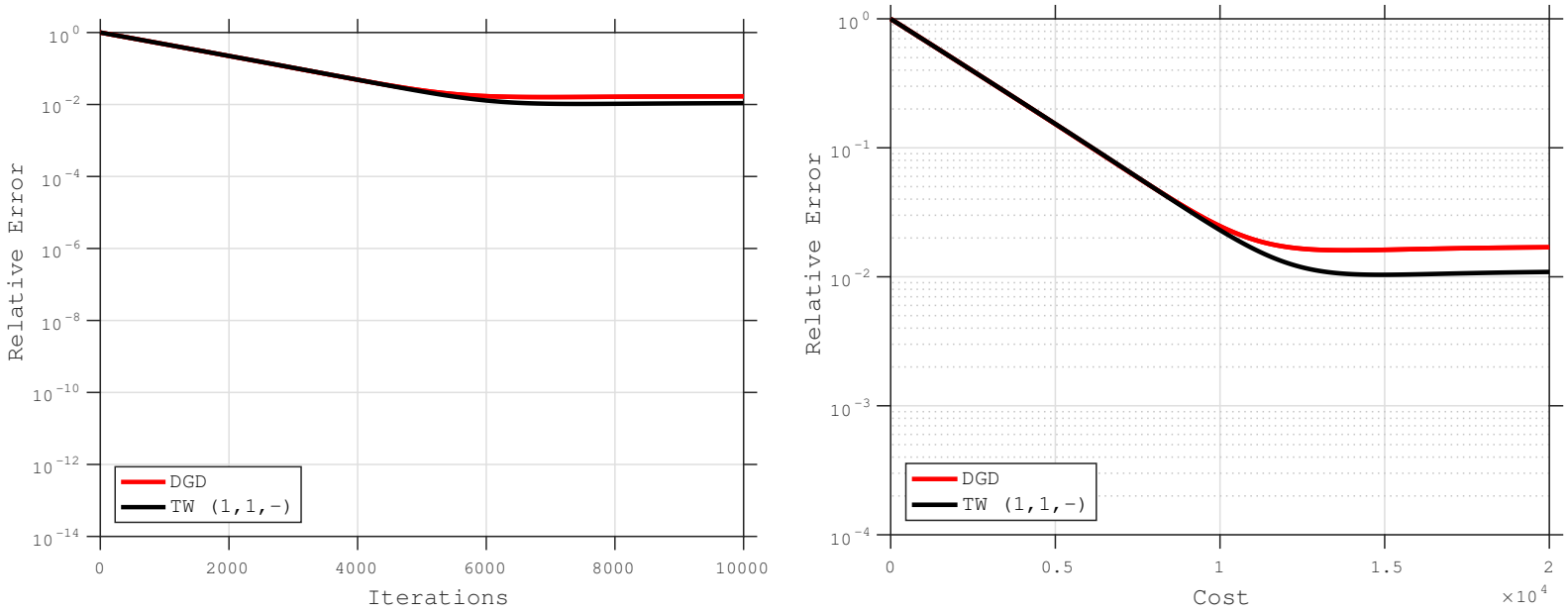
# Numerical Experiments – Quadratic Problems




Quadratic. $n = 10$, $p = 10$, $n_i = 10$, $\kappa = 10^4$.

$Cost = \#Communications \times 1 + \#Computations \times 1$

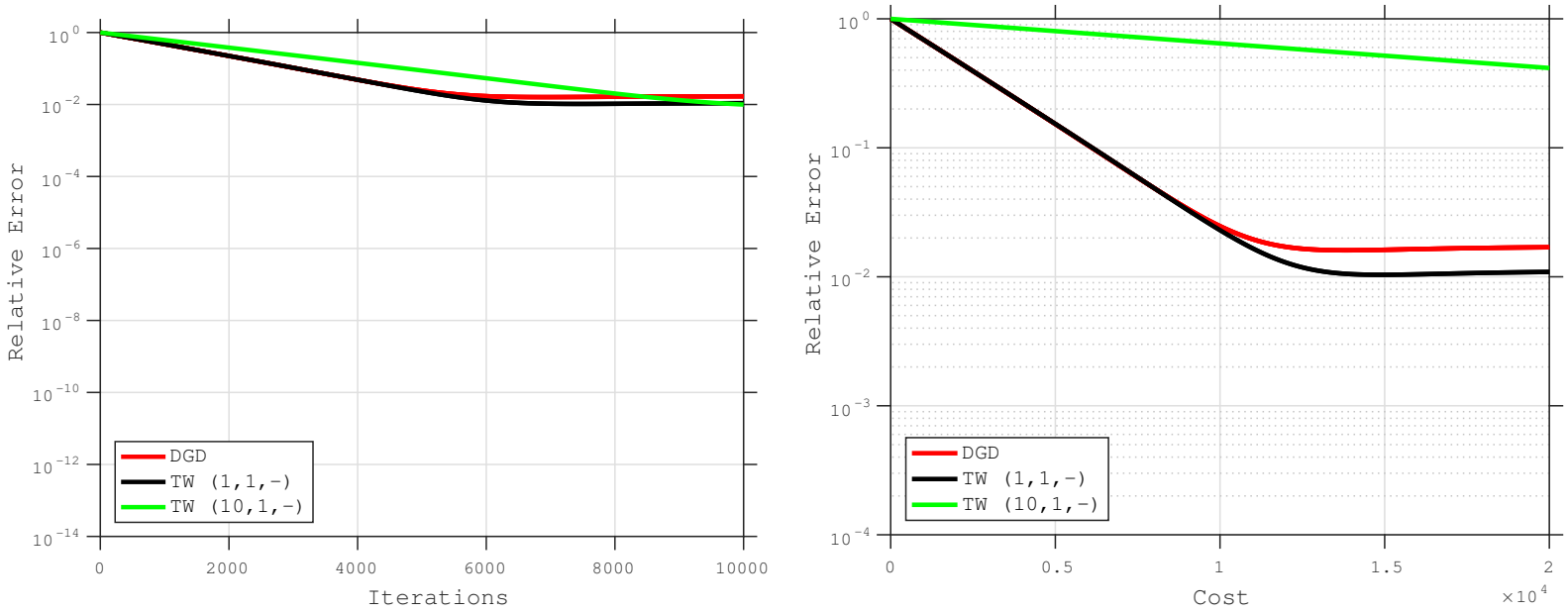
# Numerical Experiments – Quadratic Problems




Quadratic. $n = 10$, $p = 10$, $n_i = 10$, $\kappa = 10^4$.

$Cost = \#Communications \times 1 + \#Computations \times 1$

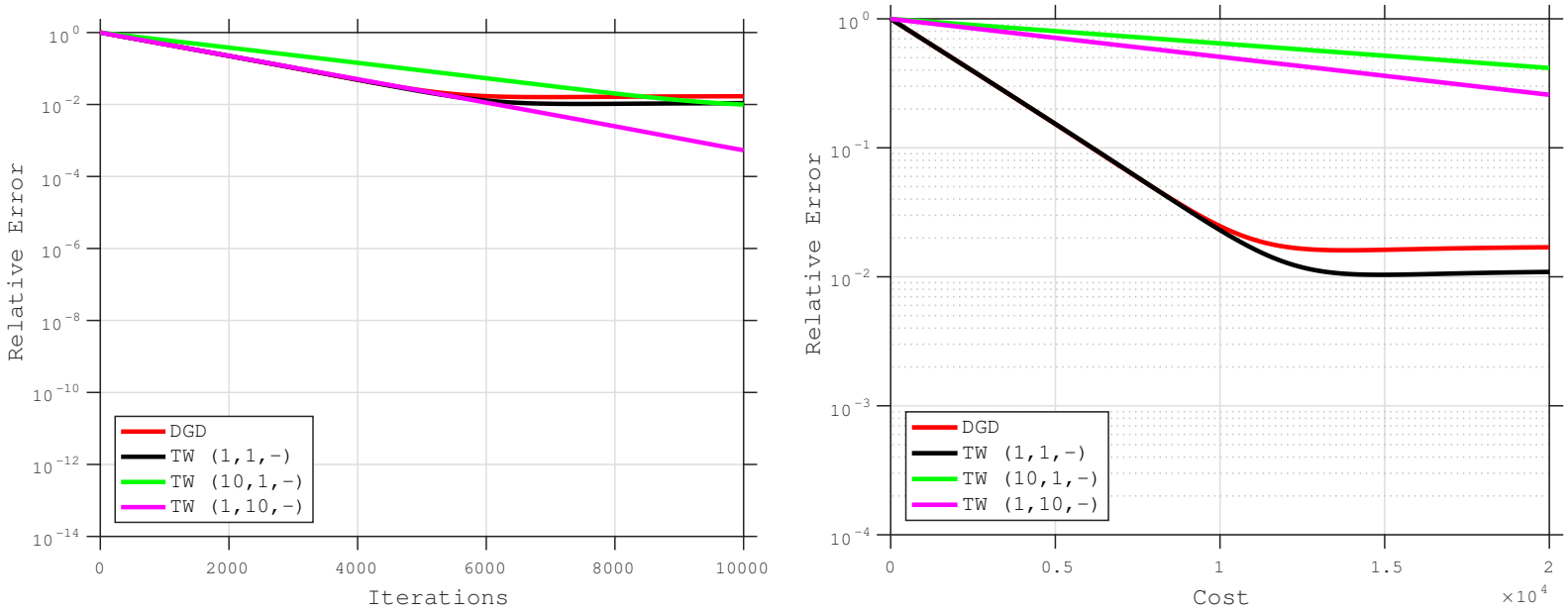
# Numerical Experiments – Quadratic Problems




Quadratic. $n = 10$, $p = 10$, $n_i = 10$, $\kappa = 10^4$.

$Cost = \#Communications \times 1 + \#Computations \times 1$

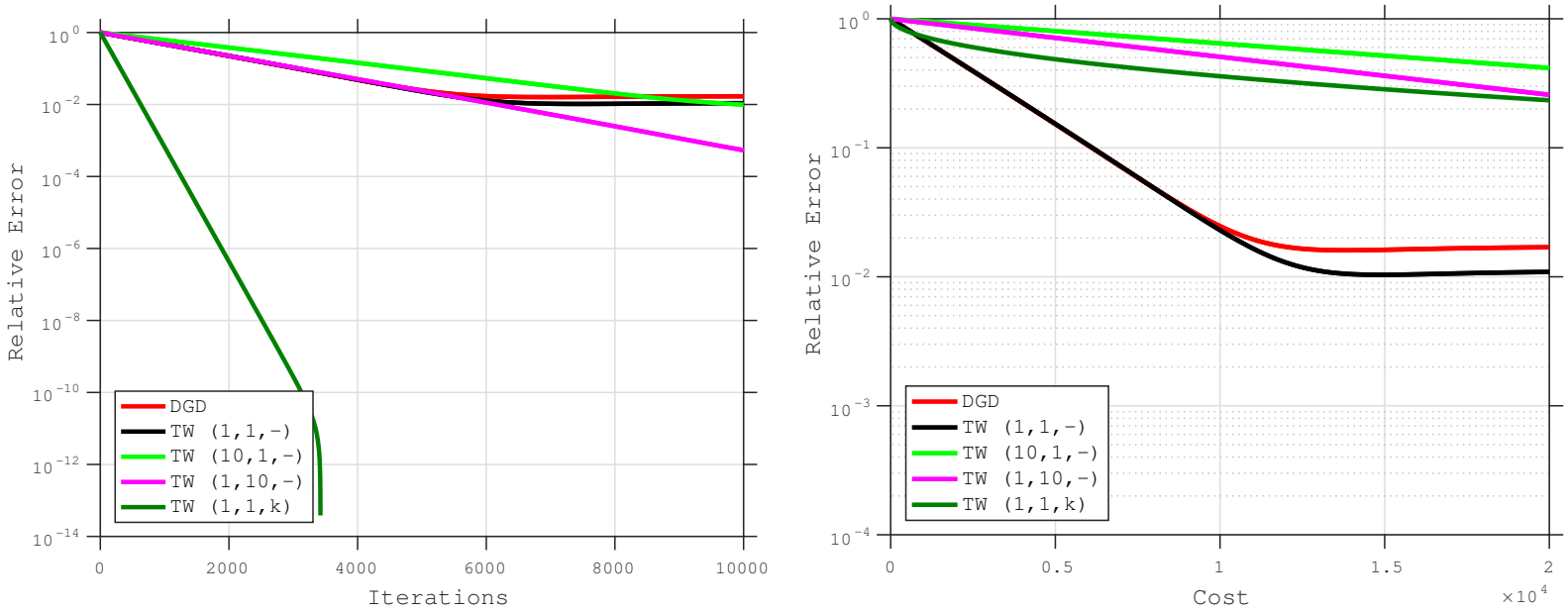
# Numerical Experiments – Quadratic Problems




Quadratic. $n = 10$, $p = 10$, $n_i = 10$, $\kappa = 10^4$.

$Cost = \#Communications \times 1 + \#Computations \times 1$

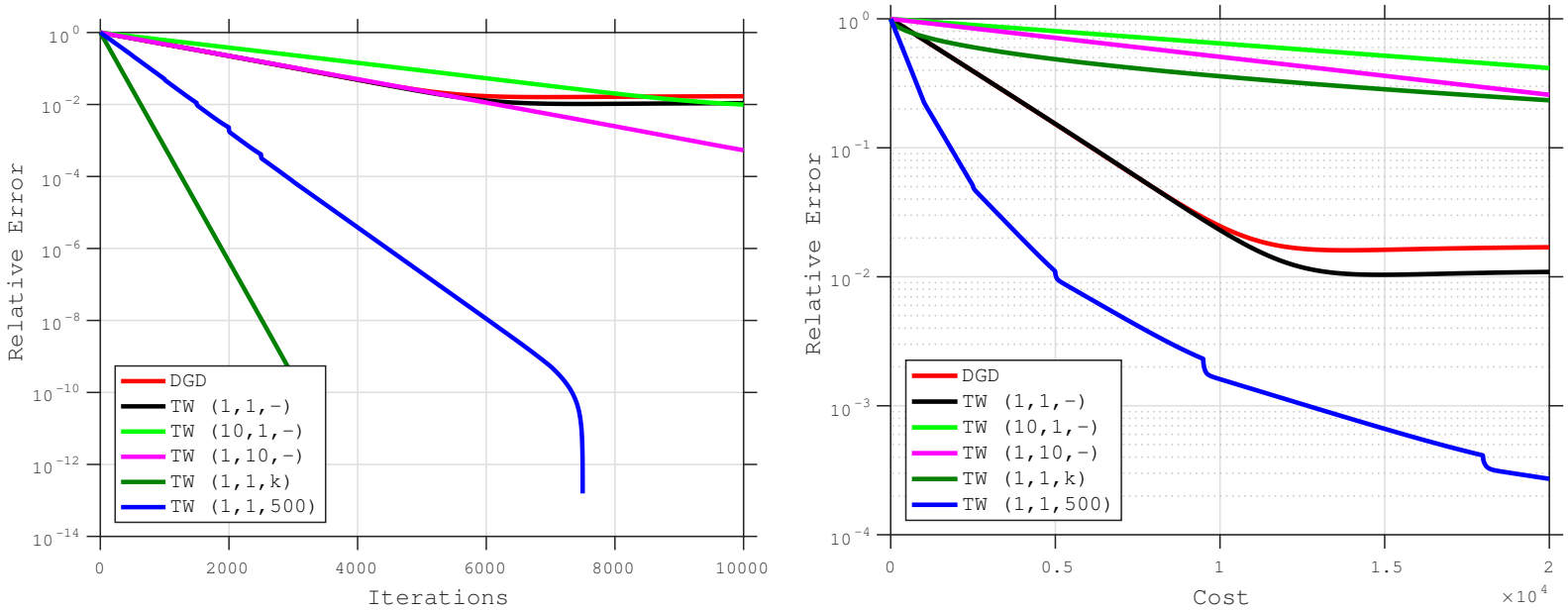
# Numerical Experiments – Quadratic Problems




Quadratic. $n = 10$, $p = 10$, $n_i = 10$, $\kappa = 10^4$.

$Cost = \#Communications \times 1 + \#Computations \times 1$

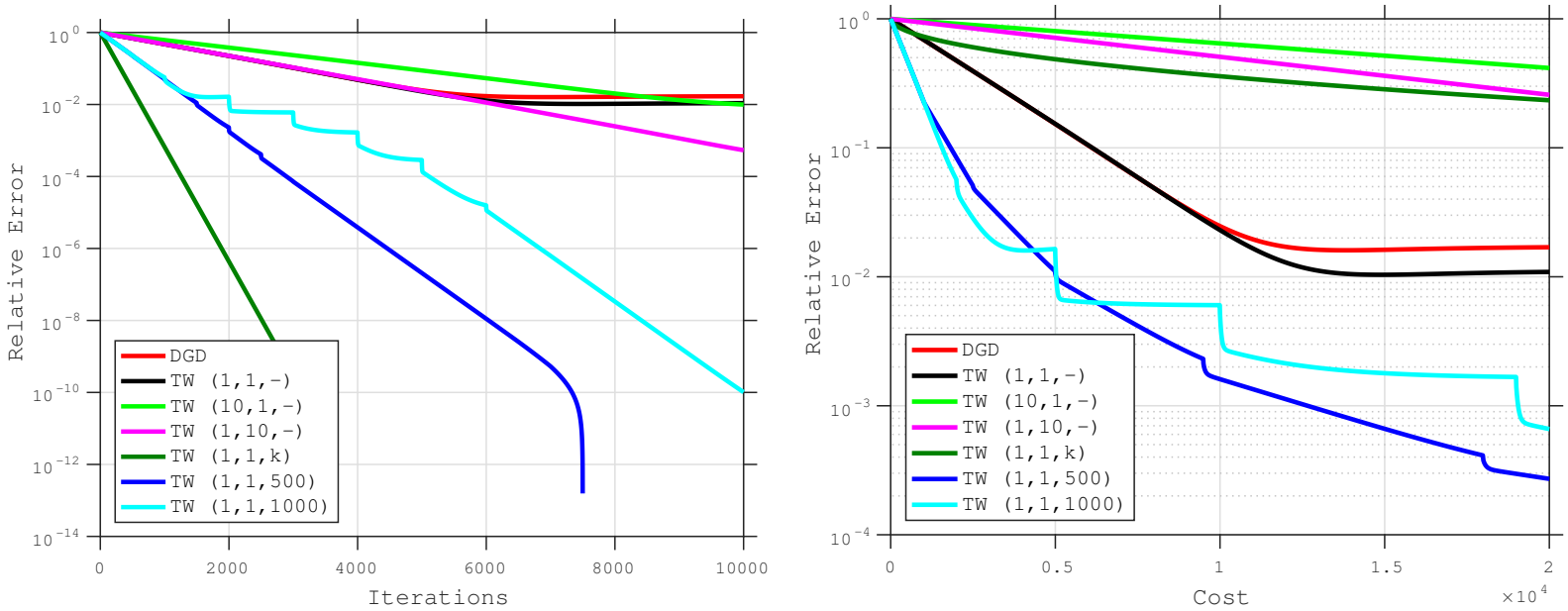
# Numerical Experiments – Quadratic Problems




Quadratic. $n = 10$, $p = 10$, $n_i = 10$, $\kappa = 10^4$.

$Cost = \#Communications \times 1 + \#Computations \times 1$

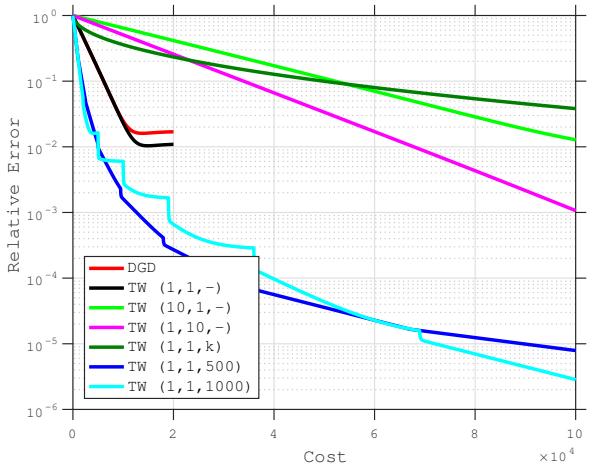
# Numerical Experiments – Quadratic Problems




Quadratic. $n = 10$, $p = 10$, $n_i = 10$, $\kappa = 10^4$.

$Cost = \#Communications \times 1 + \#Computations \times 1$

# Experiments – Quadratic Problems – Different Costs

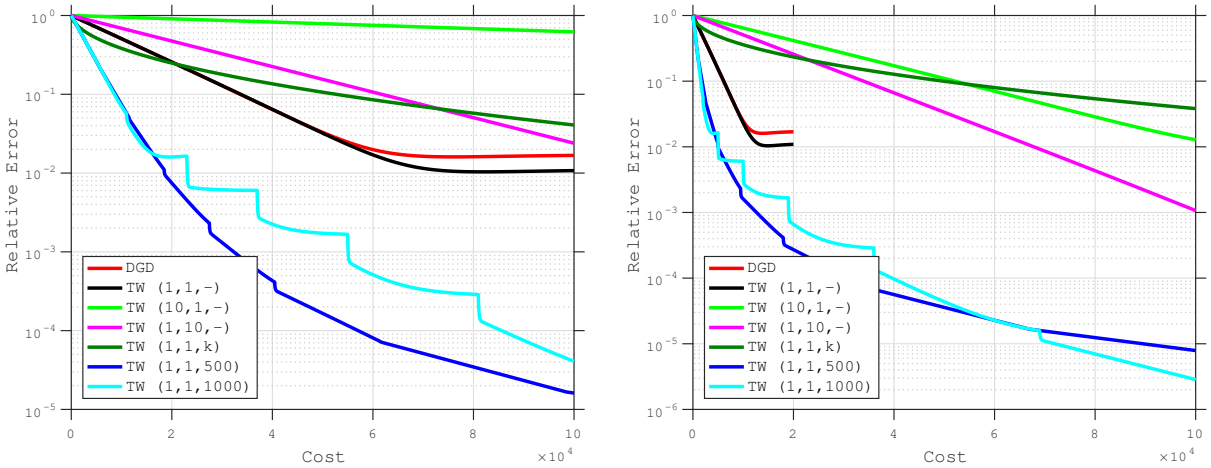



Quadratic. $n = 10$, $p = 10$, $n_i = 10$, $\kappa = 10^4$.

**Center**: $c_g = 1$, $c_c = 1$;

$$Cost = \#Communications \times c_c + \#Computations \times c_g$$

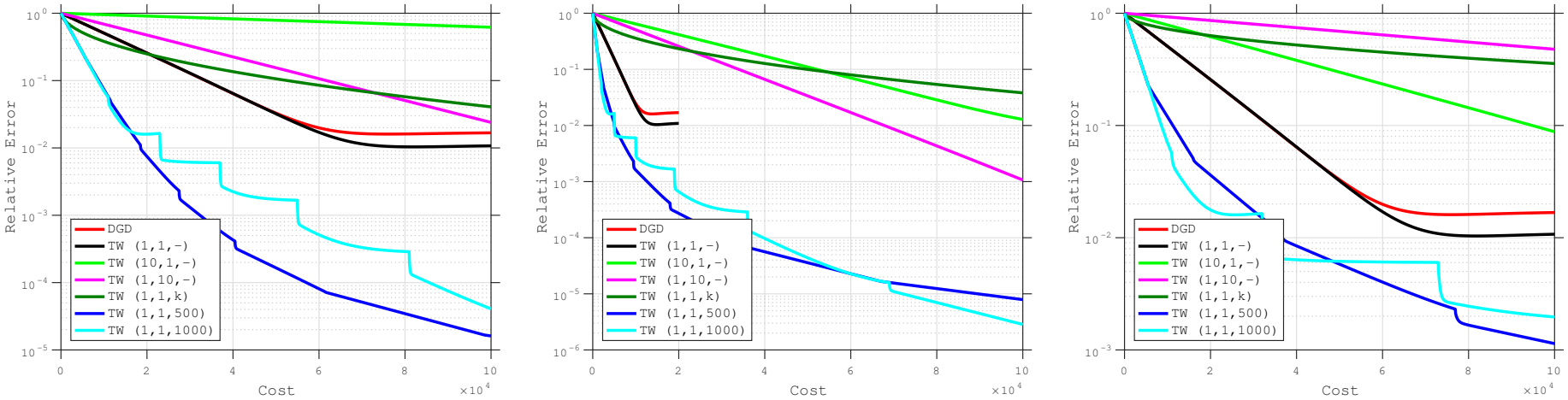
# Experiments – Quadratic Problems – Different Costs




Quadratic. $n = 10$, $p = 10$, $n_i = 10$, $\kappa = 10^4$.
**Left**: $c_g = 10$, $c_c = 1$;
**Center**: $c_g = 1$, $c_c = 1$;

$$Cost = \#Communications \times c_c + \#Computations \times c_g$$

# Experiments – Quadratic Problems – Different Costs




Quadratic. $n = 10$, $p = 10$, $n_i = 10$, $\kappa = 10^4$.
**Left**: $c_g = 10$, $c_c = 1$;
**Center**: $c_g = 1$, $c_c = 1$;
**Right**: $c_g = 1$, $c_c = 10$.

$$Cost = \#Communications \times c_c + \#Computations \times c_g$$

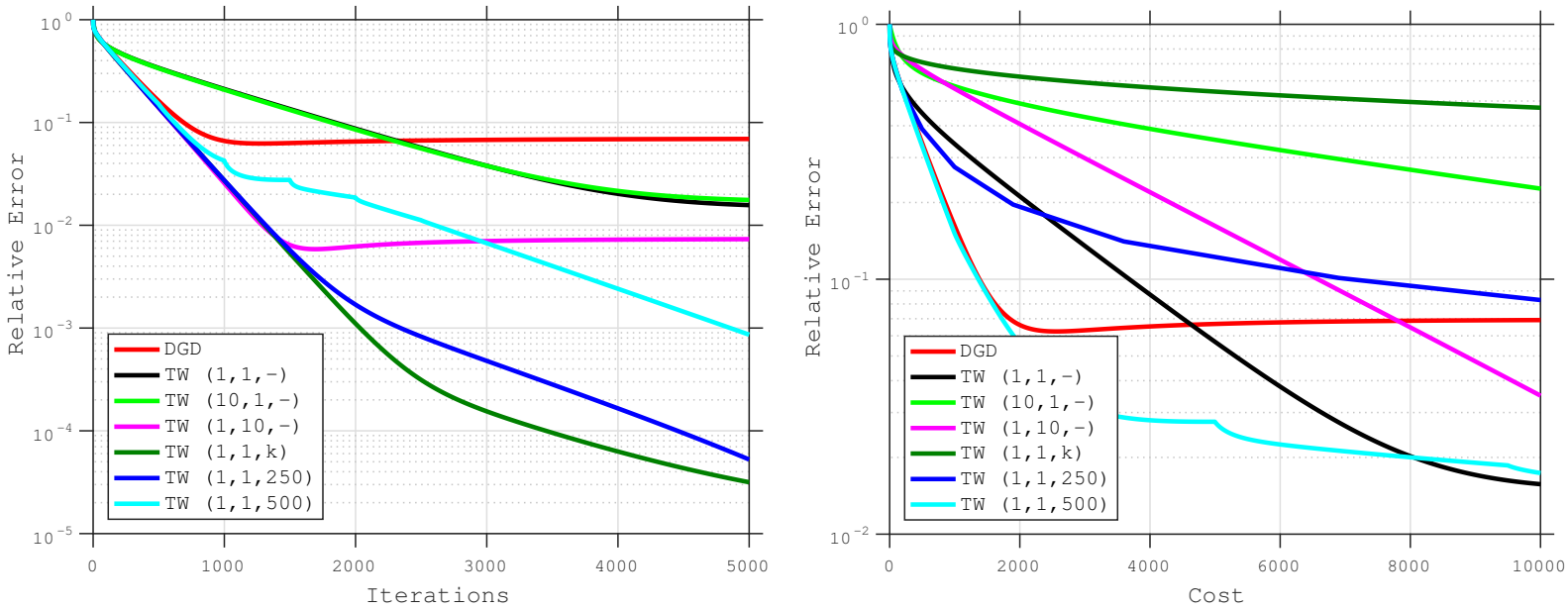
# Numerical Experiments – Logistic Regression

- **Problem**: Logistic Regression - Binary Classification (Mushroom Dataset)

$$f(x) = \frac{1}{n \cdot n_i} \sum_{i=1}^{n} \sum_{j=1}^{n_i} \log(1 + e^{-(b_i)_j (x^T (A_i)_{j\cdot})})$$

  where $A \in \mathbb{R}^{n \cdot n_i \times p}$ and $b \in \{-1, 1\}^{n \cdot n_i}$, and each node $i = 1, ..., n$ has a portion of $A$ and $b$, $A_i \in \mathbb{R}^{n_i \times p}$ and $b_i \in \mathbb{R}^{n_i}$
- **Parameters**: $n = 10$, $p = 114$, $n_i = 812$, $\kappa = 10^4$
- **Graph**: 4-cyclic graph

# Numerical Experiments – Logistic Regression




Logistic Regression - `mushroom`. $n = 10$, $p = 114$, $n_i = 812$.

$$Cost = \#Communications \times 1 + \#Computations \times 1$$

# Overview

# Final Remarks

- Most distributed optimization algorithms do one communication and one computation per iteration

# Final Remarks

- Most distributed optimization algorithms do one communication and one computation per iteration
- Showed the effect (theoretically and empirically) of doing multiple consensus in DGD

# Final Remarks

- Most distributed optimization algorithms do one communication and one computation per iteration
- Showed the effect (theoretically and empirically) of doing multiple consensus in DGD
- Proposed a variant of DGD, TW, that decouples the two operations (consensus and computation) and converges to the solution by performing multiple consensus steps

# Final Remarks

- Most distributed optimization algorithms do one communication and one computation per iteration
- Showed the effect (theoretically and empirically) of doing multiple consensus in DGD
- Proposed a variant of DGD, TW, that decouples the two operations (consensus and computation) and converges to the solution by performing multiple consensus steps

$$\text{DGD: } x_{k+1} = Zx_k - \alpha \nabla f(x_k)$$
$$\text{TW: } x_{k+1} = Zx_k - \alpha \nabla f(Zx_k)$$

# Final Remarks

- Most distributed optimization algorithms do one communication and one computation per iteration
- Showed the effect (theoretically and empirically) of doing multiple consensus in DGD
- Proposed a variant of DGD, TW, that decouples the two operations (consensus and computation) and converges to the solution by performing multiple consensus steps
- Important to balance communication and computation in order to get best performance in terms of cost — right balance depends on the application (e.g., cost of communication and cost of computation)

# Future Work

- Apply framework to other algorithms (exact, second-order, asynchronous, ...)
- Construct framework to do multiple gradient steps
- Adapt number of gradient and communication steps in algorithmic way
- Other considerations: memory access, partial blocks, quantization effects, dynamic environment

# Backup Slides

# DGD – Theory – Bounded gradients

## Lemma (Bounded gradients) [Yuan et. al., 2015]

Suppose Assumption 1 holds, and let the step size satisfy

$$\alpha \leq \frac{1 + \lambda_n(\mathbf{W})}{L}$$

where $\lambda_n(\mathbf{W})$ is the smallest eigenvalue of $\mathbf{W}$ and $L = \max_i L_i$. Then, starting from $x_{i,0} = 0$ $(i = 1, 2, ..., n)$, the sequence $x_{i,k}$ generated by DGD converges. In addition, we also have

$$\|\nabla \mathbf{f}(\mathbf{x}_k)\| \leq D = \sqrt{2L(\frac{1}{n}\sum_{i=1}^{n} f_i(0) - f_i^\star)} \tag{1}$$

for all $k = 1, 2, ...,$ where $f_i^\star = f_i(x_i^\star)$ and $x_i^\star = \arg\min_x f_i(x)$.

# DGD$^t$ – Theory – Bounded gradients

## Lemma (Bounded gradients) [**ASB**, RB, NSK and EW, 2017]

Suppose Assumption 1 holds, and let the step size satisfy

$$\alpha \leq \frac{1 + \lambda_n(\mathbf{W}^t)}{L}$$

where $\lambda_n(\mathbf{W}^t)$ is the smallest eigenvalue of $\mathbf{W}^t$ and $L = \max_i L_i$. Then, starting from $x_{i,0} = 0$ $(i = 1, 2, ..., n)$, the sequence $x_{i,k}$ generated by DGD converges. In addition, we also have

$$\|\nabla \mathbf{f}(\mathbf{x}_k)\| \leq D = \sqrt{2L(\frac{1}{n}\sum_{i=1}^{n} f_i(0) - f_i^\star)} \tag{1}$$

for all $k = 1, 2, ...,$ where $f_i^\star = f_i(x_i^\star)$ and $x_i^\star = \arg\min_x f_i(x)$.

# DGD – Theory – Bounded deviation from mean

## Lemma (Bounded deviation from mean) [Yuan et. al., 2015]

If (1) and Assumption 2 hold, then the total deviation from the mean is bounded, namely,

$$\|x_{i,k} - \bar{x}_k\| \leq \frac{\alpha D}{1-\beta}$$

for all $k$ and $i$. Moreover, if in addition Assumption 1 holds, then

$$\|\nabla f_i(x_{i,k}) - \nabla f_i(\bar{x}_k)\| \leq \frac{\alpha D L_i}{1-\beta}$$

$$\|\nabla f(x_k) - \nabla f(\bar{x}_k)\| \leq \frac{\alpha D L}{1-\beta}$$

for all $k$ and $i$.

# DGD$^t$ – Theory – Bounded deviation from mean

## Lemma (Bounded deviation from mean) [**ASB**, RB, NSK and EW, 2017]

If (1) and Assumption 2 hold, then the total deviation from the mean is bounded, namely,

$$\|x_{i,k} - \bar{x}_k\| \leq \frac{\alpha D}{1 - \beta^t}$$

for all $k$ and $i$. Moreover, if in addition Assumption 1 holds, then

$$\|\nabla f_i(x_{i,k}) - \nabla f_i(\bar{x}_k)\| \leq \frac{\alpha D L_i}{1 - \beta^t}$$

$$\|\nabla f(x_k) - \nabla f(\bar{x}_k)\| \leq \frac{\alpha D L}{1 - \beta^t}$$

for all $k$ and $i$.

# $TW^t$ – Theory – Bounded deviation from mean

## Lemma (Bounded deviation from mean) [**ASB**, RB, NSK and EW, 2017]

If Assumption 2 holds, then the total deviation from the mean is bounded, namely,

$$\|y_{i,k} - \bar{x}_k\| \leq \frac{\beta^t \alpha D k(k+1)}{2}$$

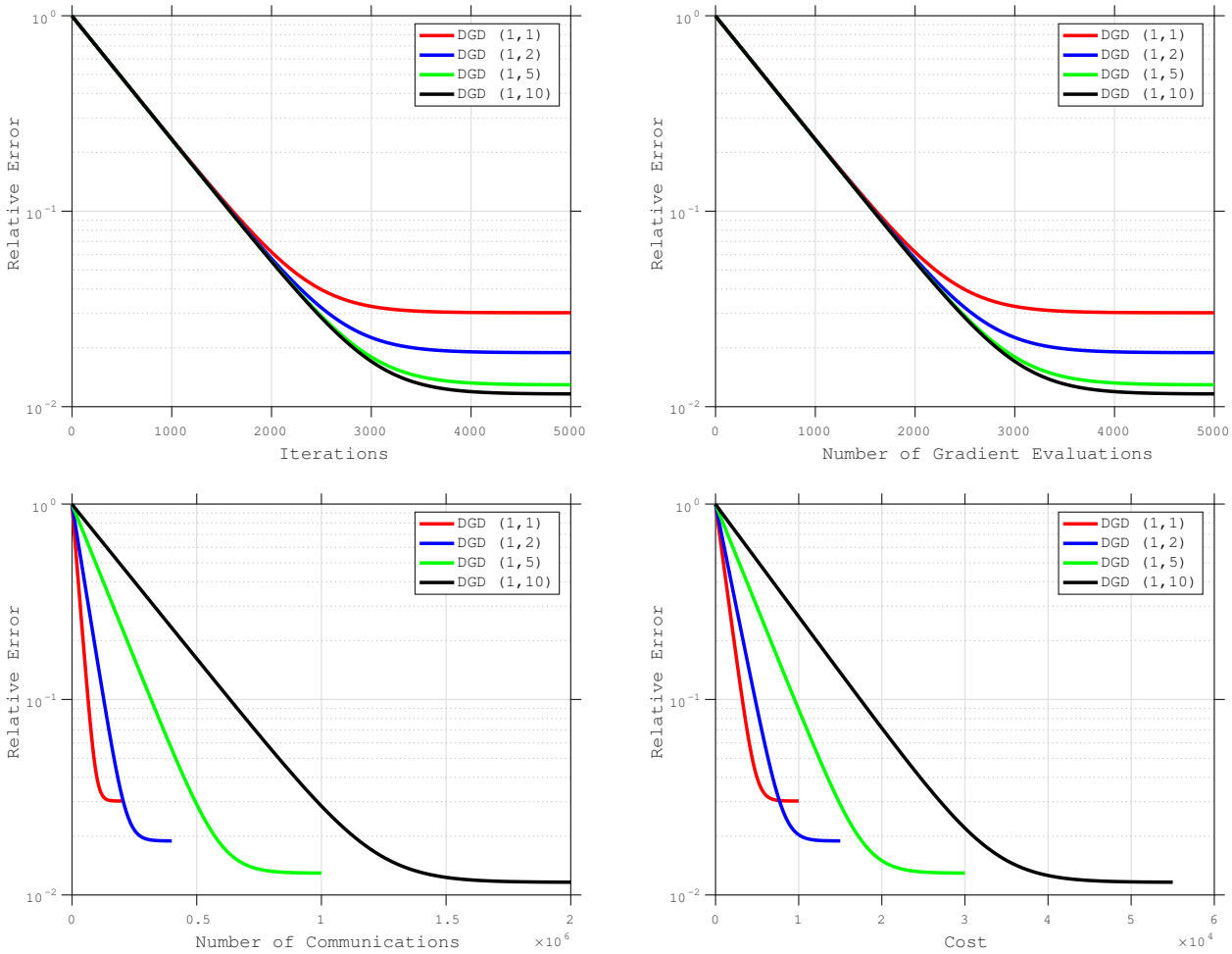
for all $k$ and $i$. Moreover, if in addition Assumption 1 holds, then

$$\|\nabla f_i(y_{i,k}) - \nabla f_i(\bar{x}_k)\| \leq \frac{\beta^t \alpha D L_i k(k+1)}{2}$$

$$\|\nabla f(y_k) - \nabla f(\bar{x}_k)\| \leq \frac{\beta^t \alpha D L k(k+1)}{2}$$

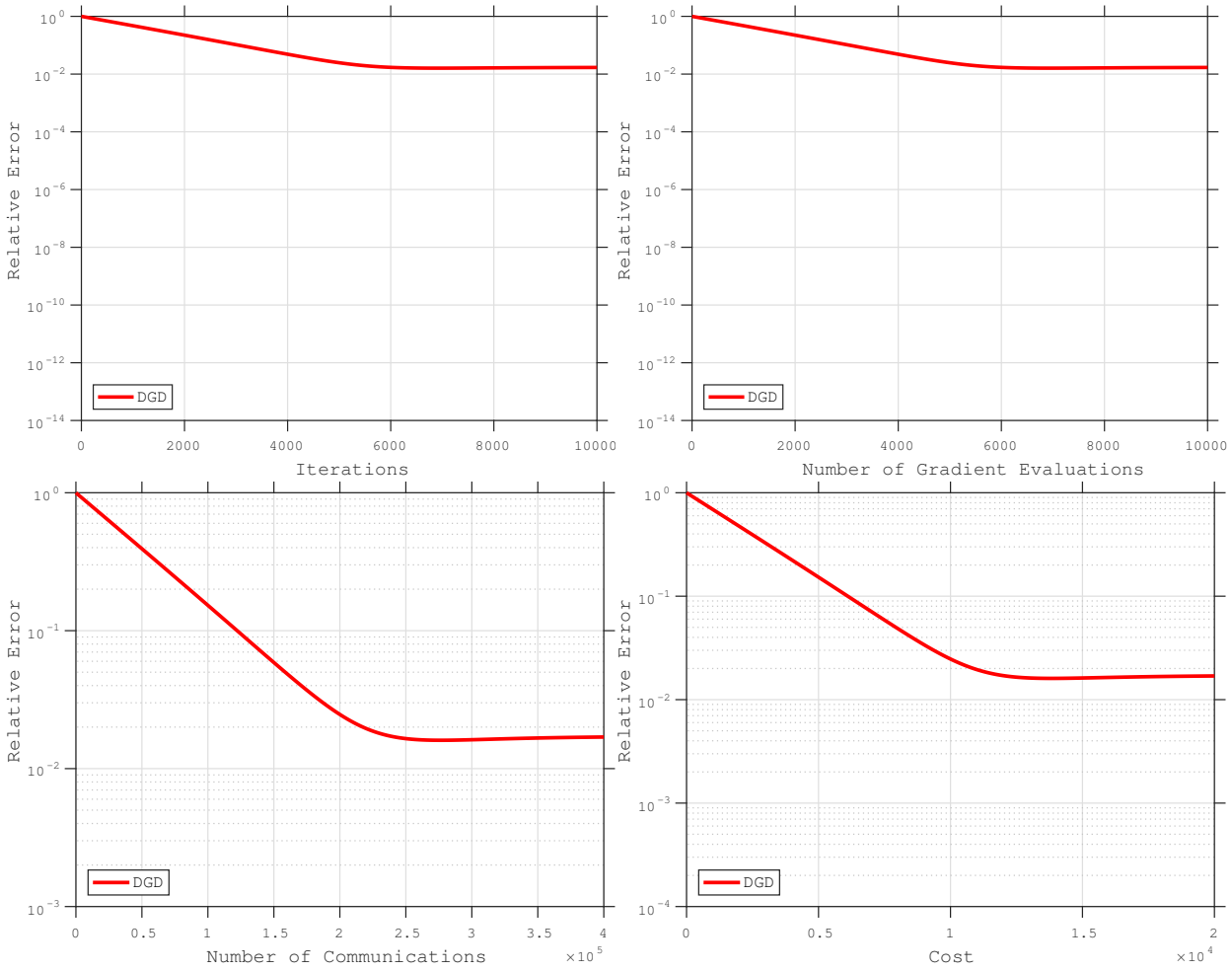
for all $k$ and $i$.

# DGD$^t$ Numerical Results



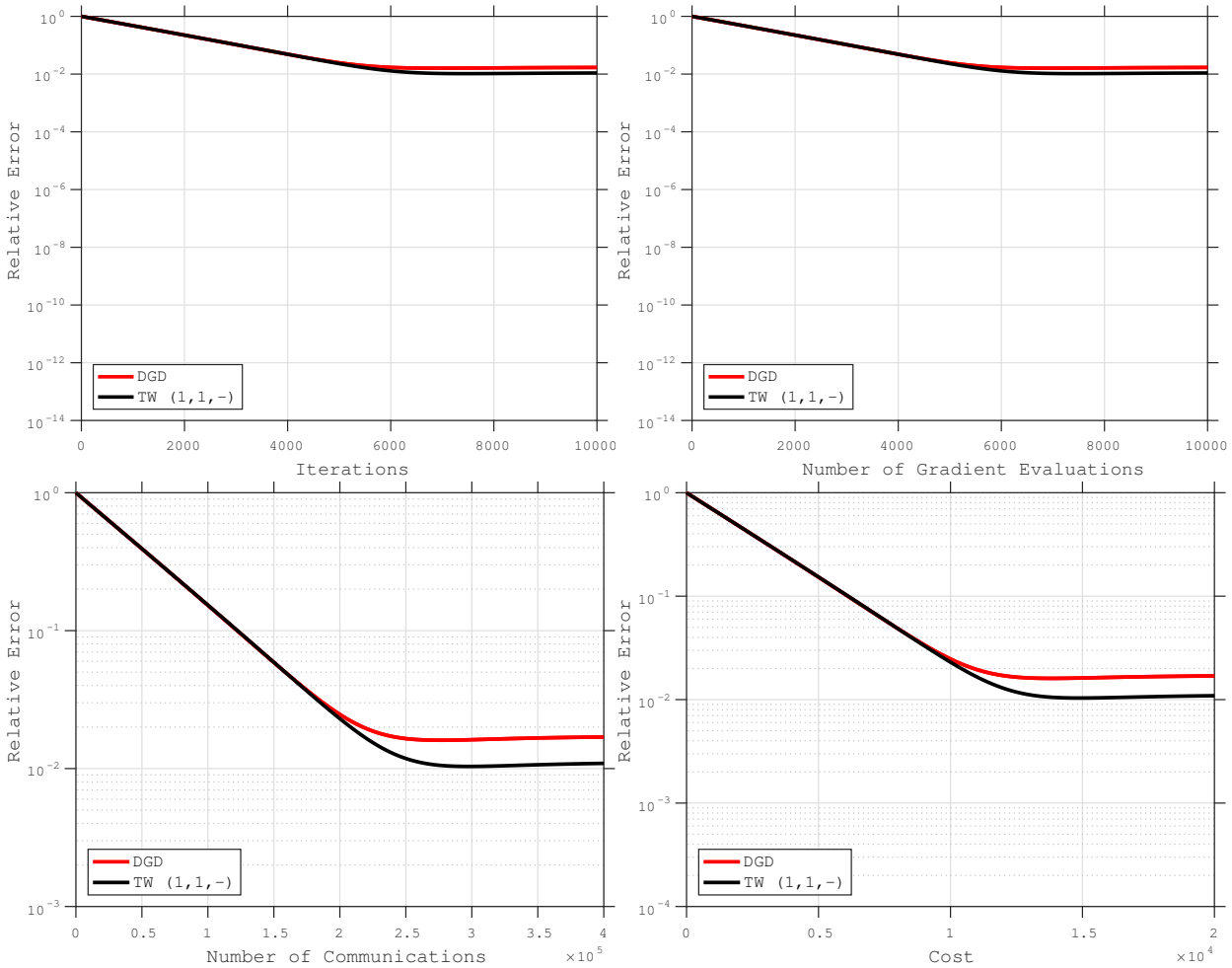

Quadratic. $n = 10$, $p = 10$, $n_i = 10$, $\kappa = 10^2$.

# Experiments – Quadratic Problems



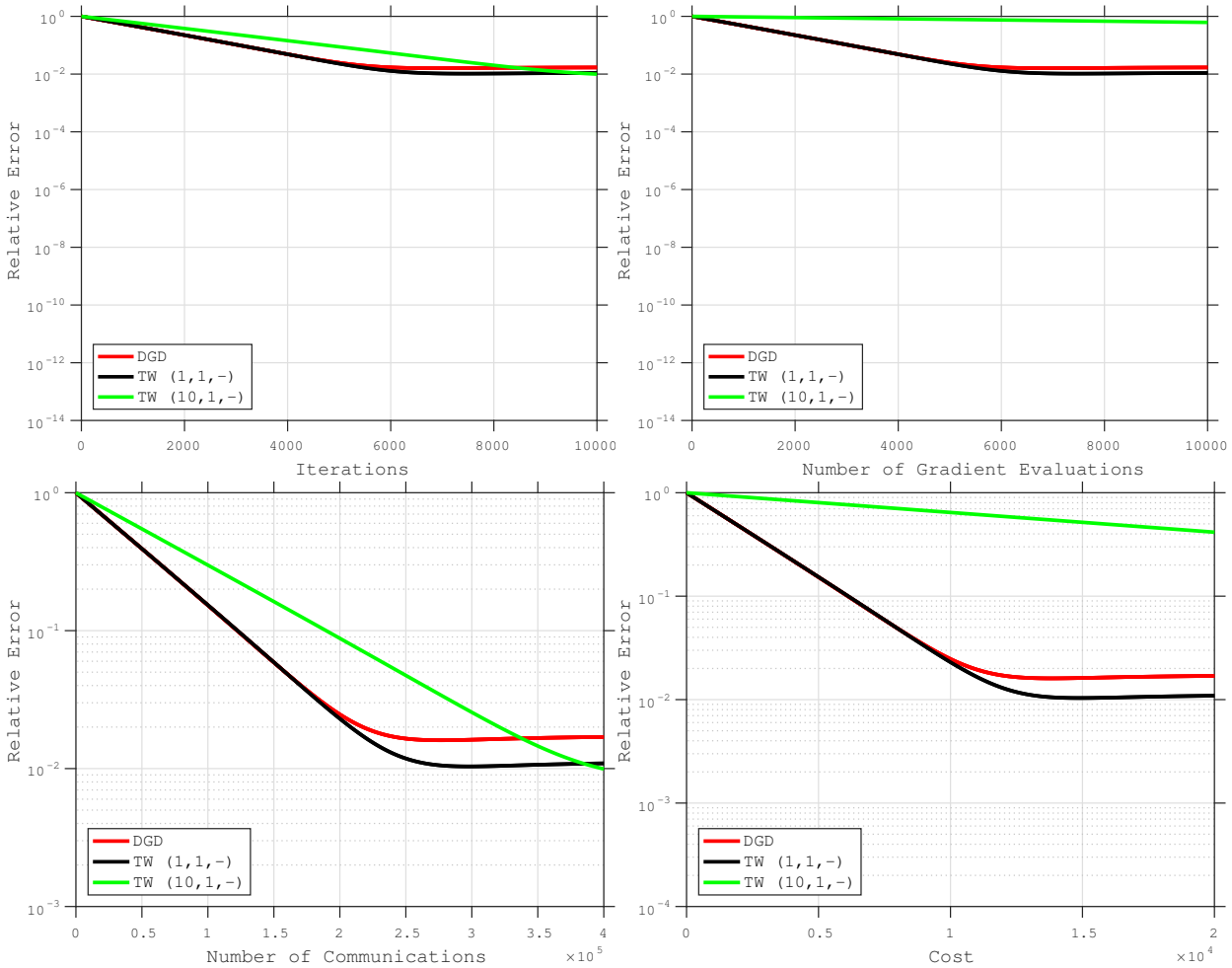

Quadratic. $n = 10$, $d = 10$, $n_i = 10$, $\kappa = 10^4$.

# Experiments – Quadratic Problems



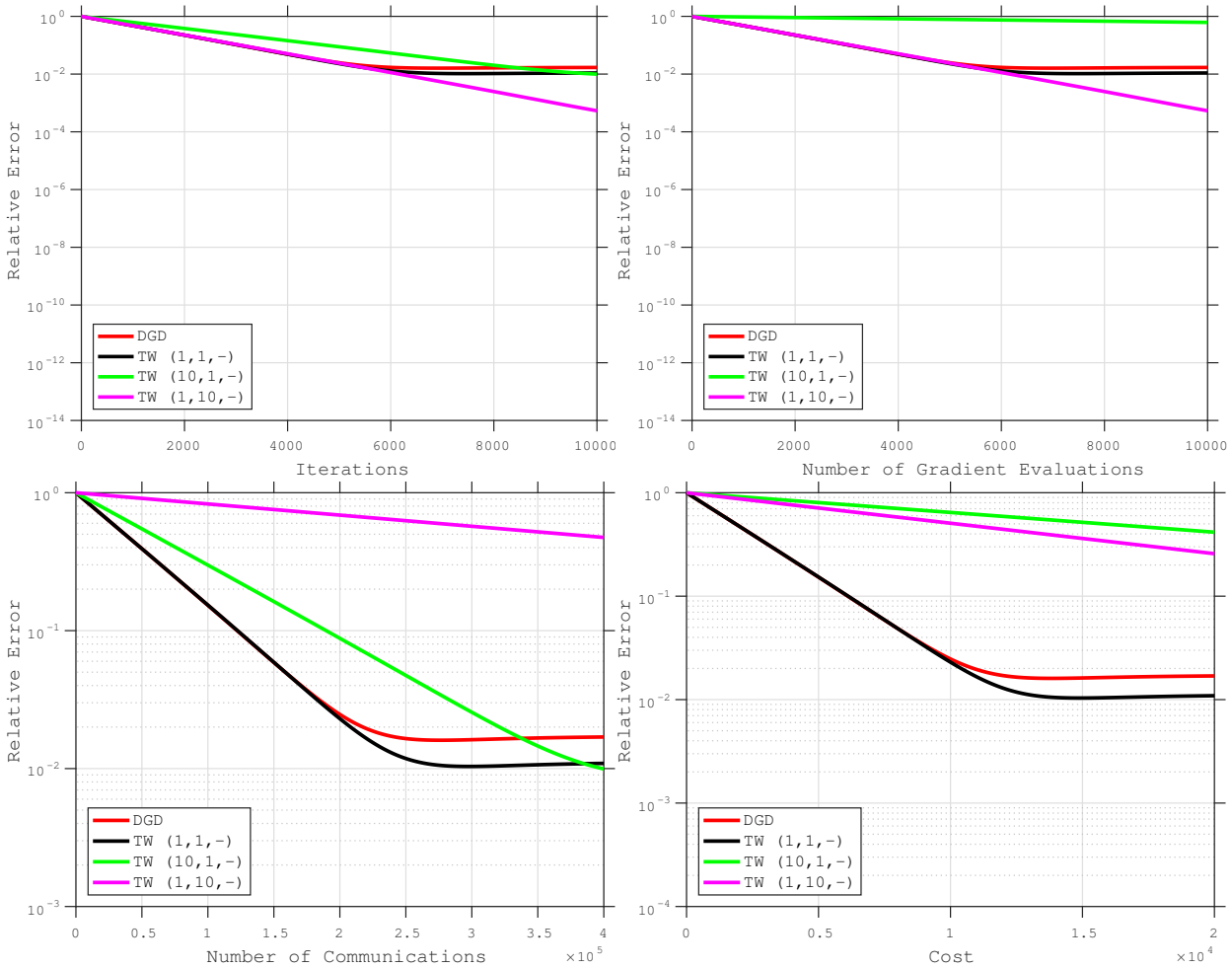

Quadratic. $n = 10$, $d = 10$, $n_i = 10$, $\kappa = 10^4$.

# Experiments – Quadratic Problems



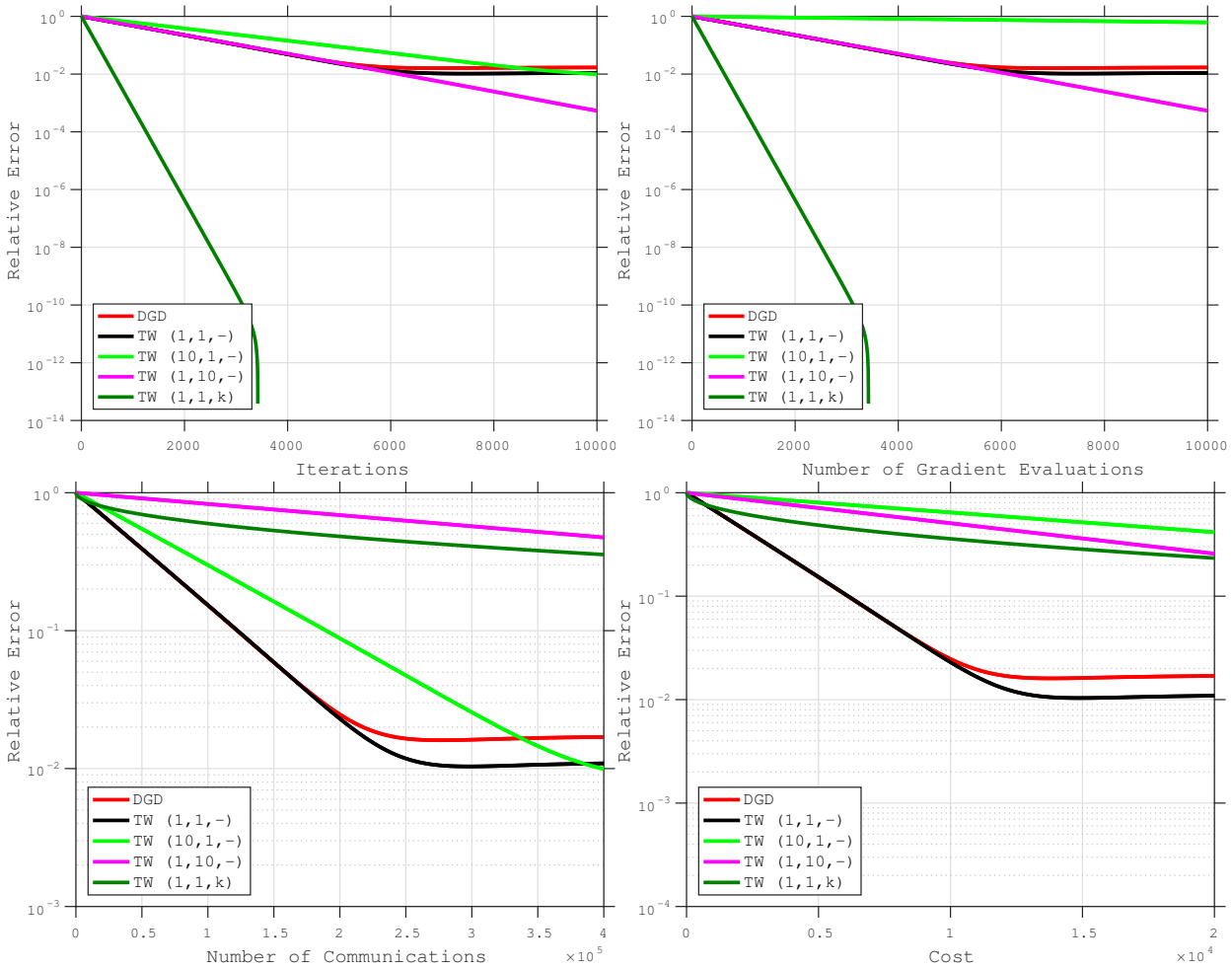

Quadratic. $n = 10$, $d = 10$, $n_i = 10$, $\kappa = 10^4$.

# Experiments – Quadratic Problems



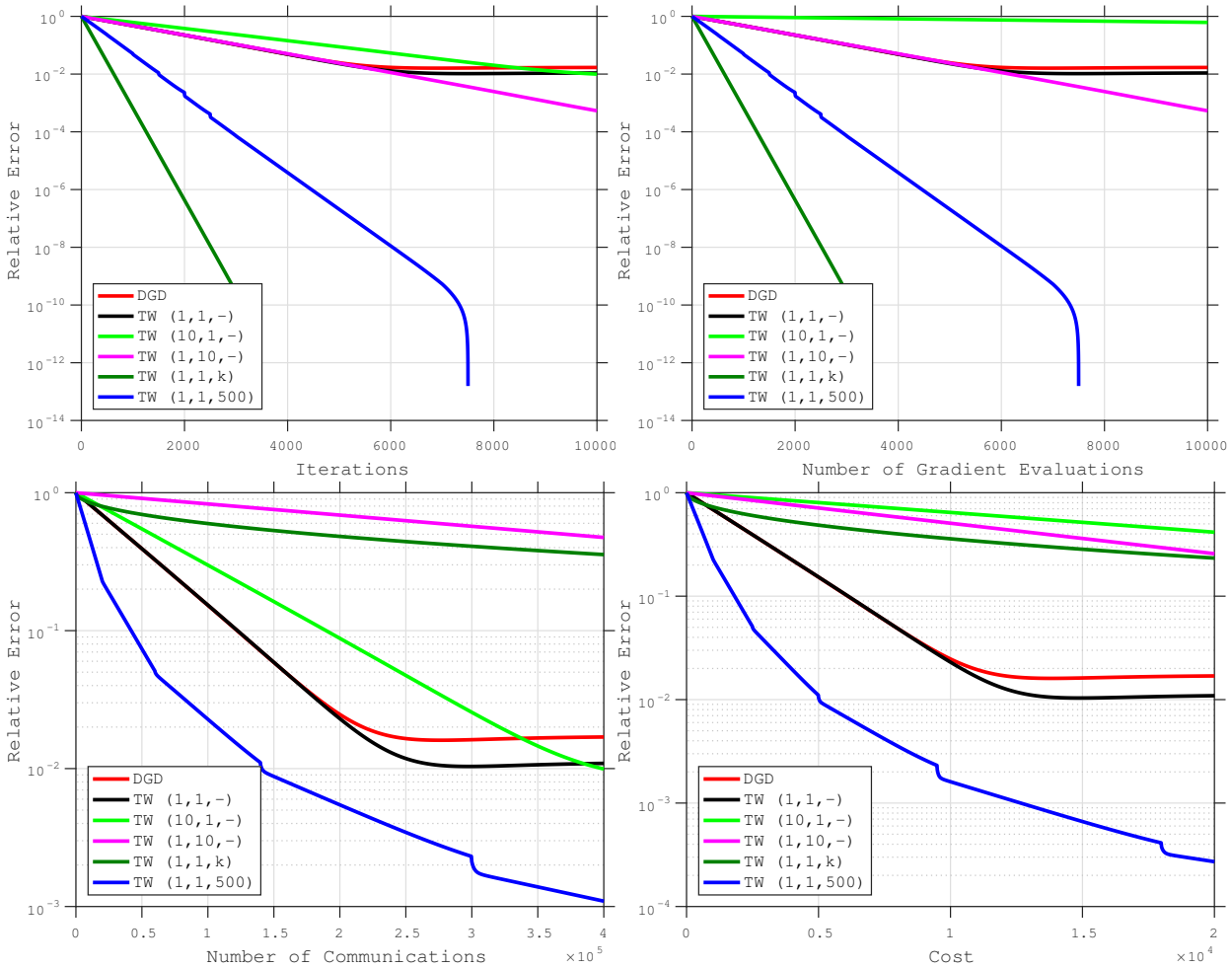

Quadratic. $n = 10$, $d = 10$, $n_i = 10$, $\kappa = 10^4$.

# Experiments – Quadratic Problems



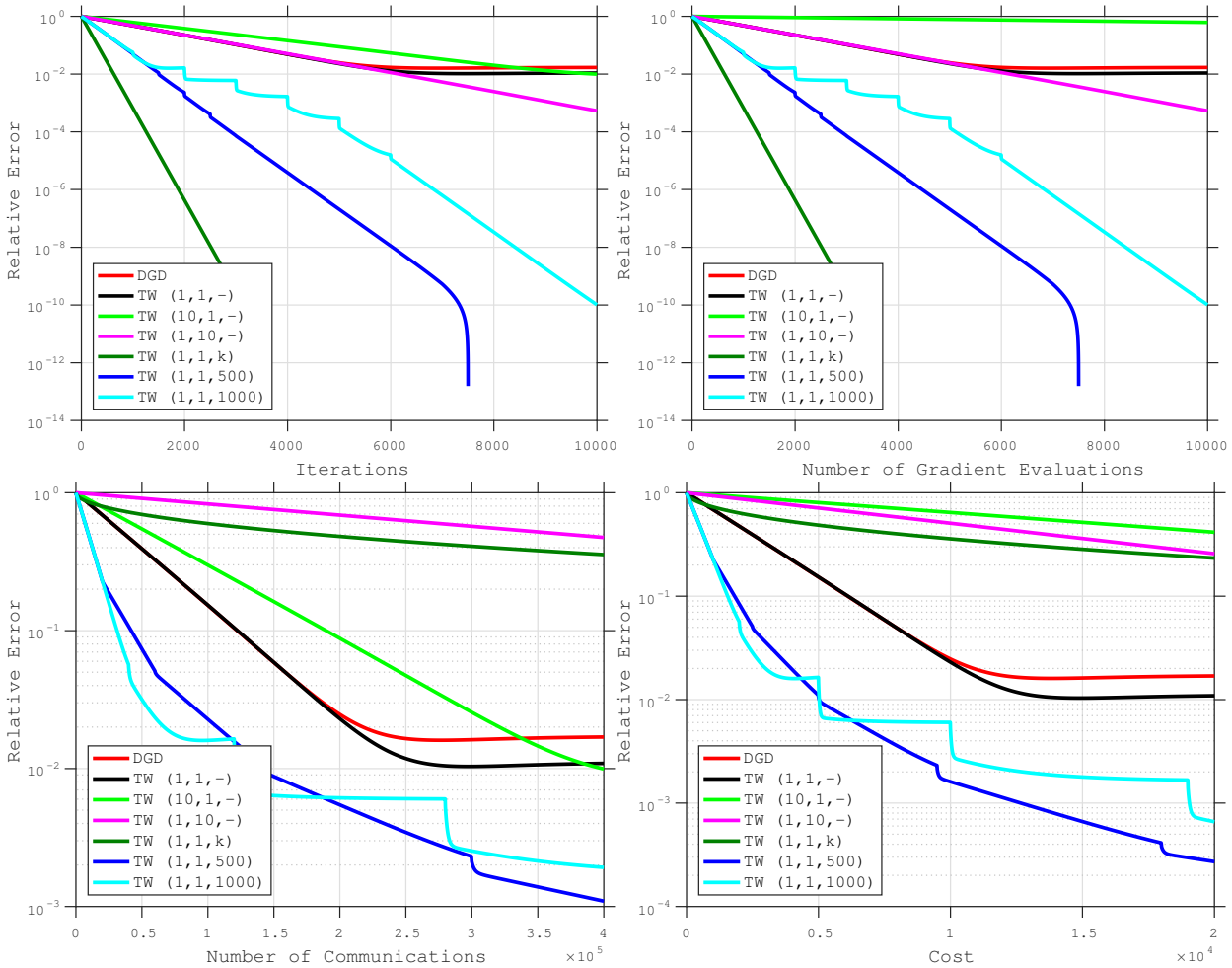

Quadratic. $n = 10$, $d = 10$, $n_i = 10$, $\kappa = 10^4$.

# Experiments – Quadratic Problems



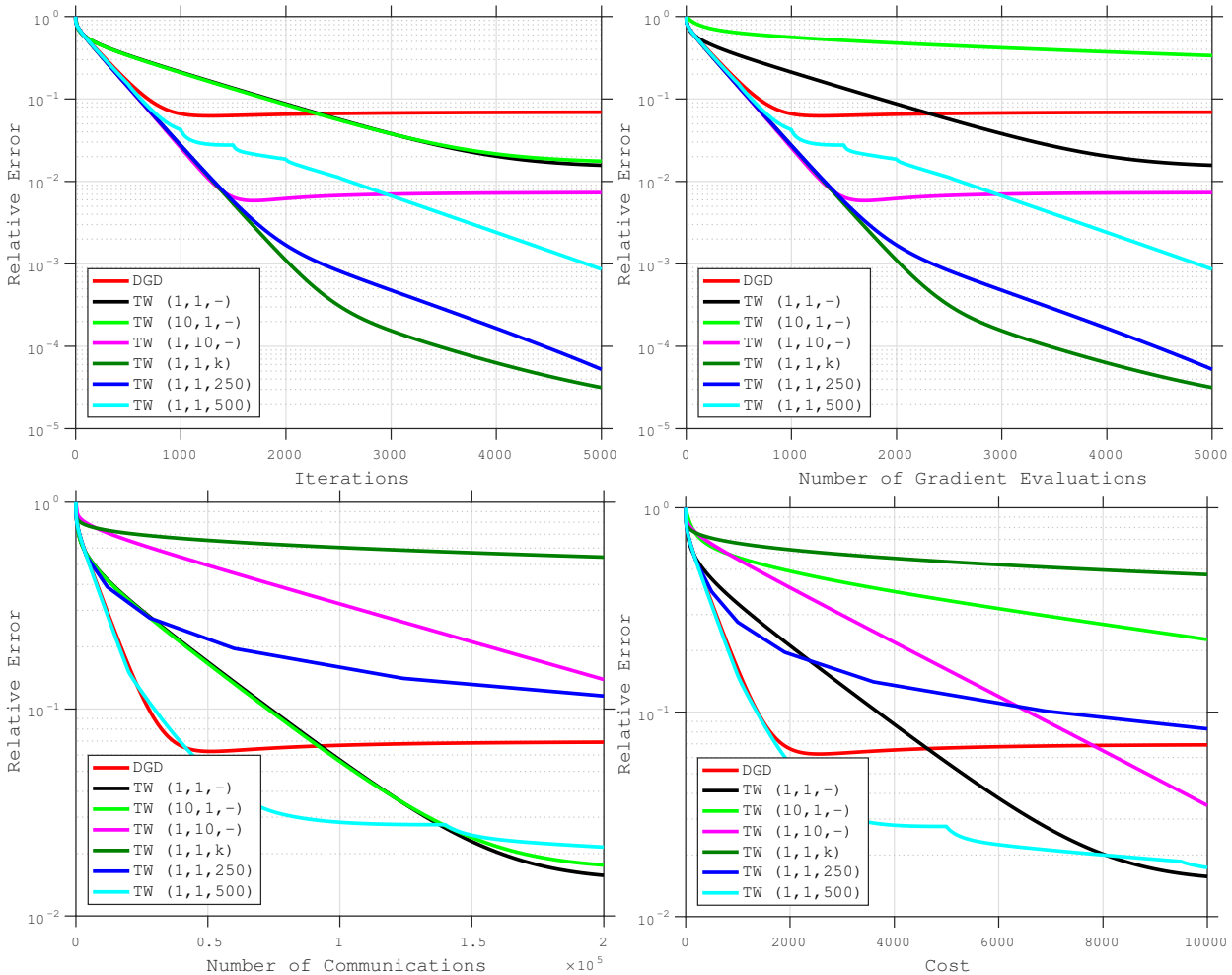

Quadratic. $n = 10$, $d = 10$, $n_i = 10$, $\kappa = 10^4$.

# Experiments – Quadratic Problems




Quadratic. $n = 10$, $d = 10$, $n_i = 10$, $\kappa = 10^4$.

# Experiments – Logistic Regression




Logistic Regression - `mushroom`. $n = 10$, $d = 114$, $n_i = 812$.