Decentralized Algorithms for Spatially Distributed Systems

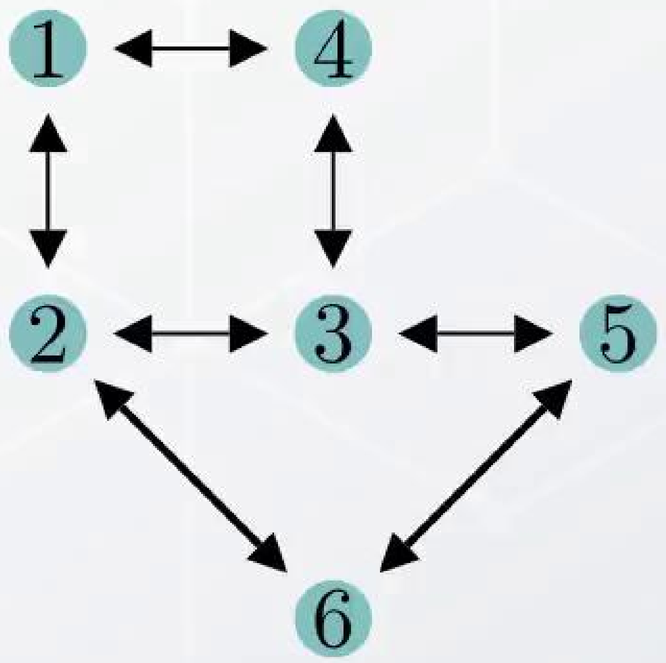
Guohui Song

Old Dominion University

Joint work with Nazar Emirov and Qiyu Sun

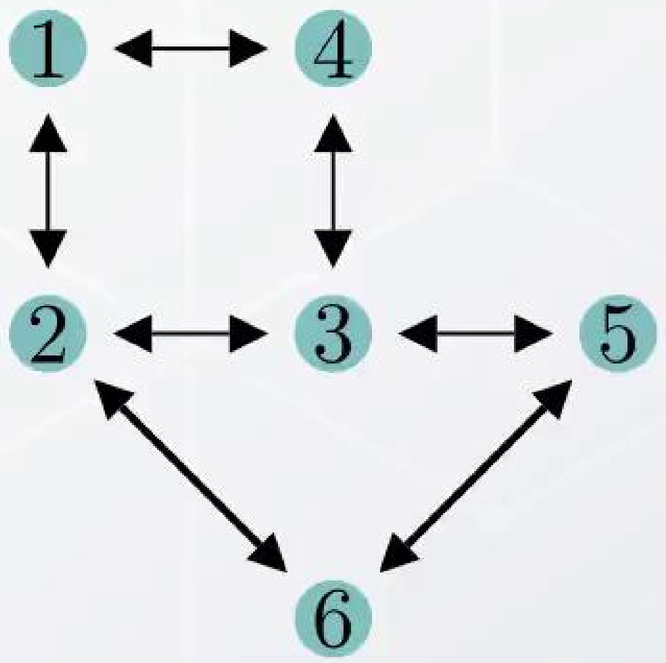
Decentralized Optimization on Graphs

- Multiple nodes/agents with computation, communication, and storage.
- Each agent would perform its own computation at each iteration.
- No central coordinator.
- Each agent would communicate with its neighbors at each iteration.
- Graphs would be used to model the communication topology.



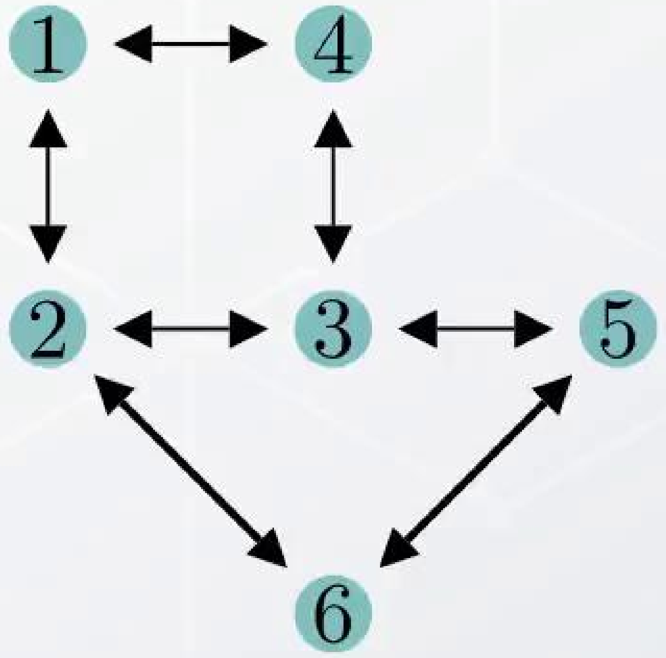
Decentralized Optimization on Graphs

- [x] Multiple nodes/agents with computation, communication, and storage.
- [ ] Each agent would perform its own computation at each iteration.
- [ ] No central coordinator.
- [ ] Each agent would communicate with its neighbors at each iteration.
- [ ] Graphs would be used to model the communication topology.



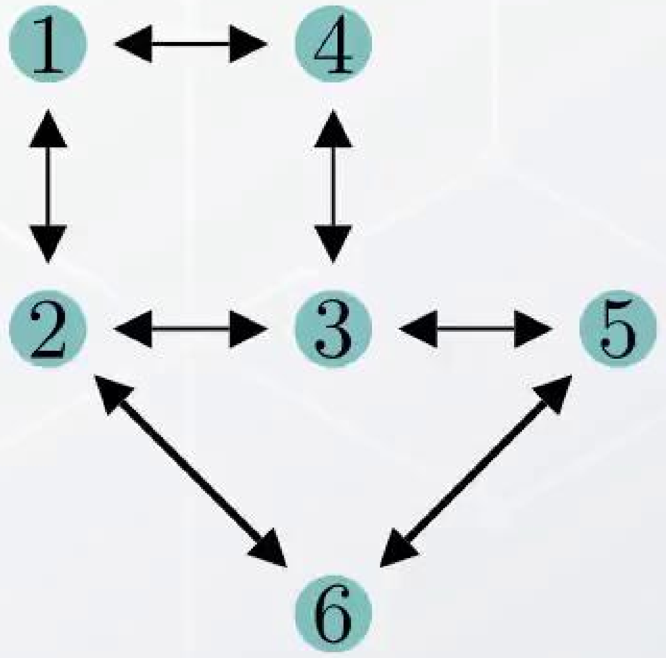
Decentralized Optimization on Graphs

- ☑ Multiple nodes/agents with computation, communication, and storage.
- ☑ Each agent would perform its own computation at each iteration.
- ☐ No central coordinator.
- ☐ Each agent would communicate with its neighbors at each iteration.
- ☐ Graphs would be used to model the communication topology.



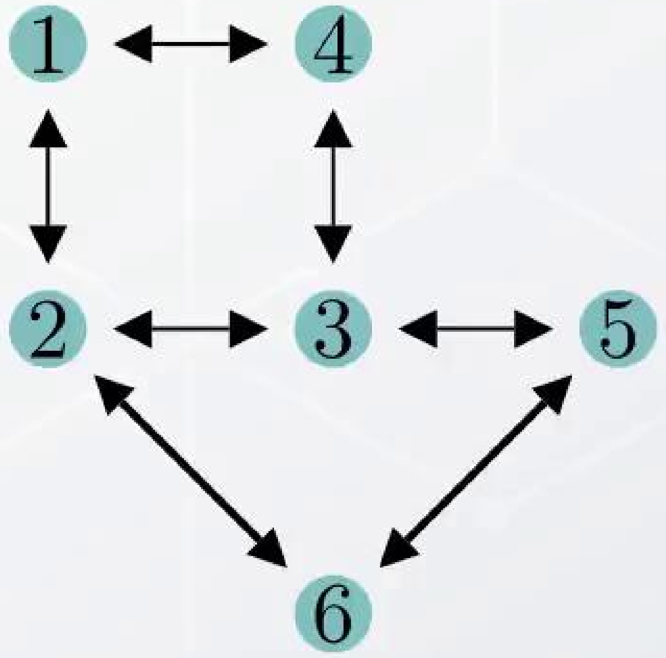
Decentralized Optimization on Graphs

- ☑ Multiple nodes/agents with computation, communication, and storage.
- ☑ Each agent would perform its own computation at each iteration.
- ☑ No central coordinator.
- ☐ Each agent would communicate with its neighbors at each iteration.
- ☐ Graphs would be used to model the communication topology.



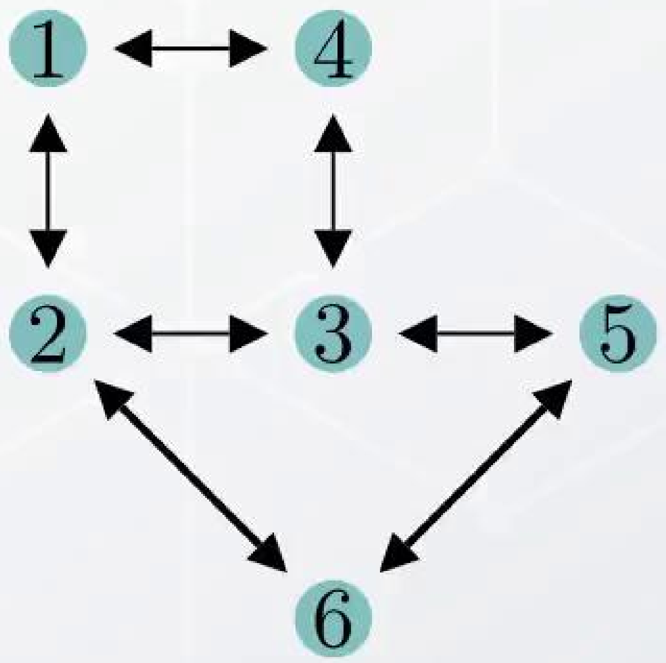
Decentralized Optimization on Graphs

- ☑ Multiple nodes/agents with computation, communication, and storage.
- ☑ Each agent would perform its own computation at each iteration.
- ☑ No central coordinator.
- ☑ Each agent would communicate with its neighbors at each iteration.
- ☐ Graphs would be used to model the communication topology.



Decentralized Optimization on Graphs

- ☑ Multiple nodes/agents with computation, communication, and storage.
- ☑ Each agent would perform its own computation at each iteration.
- ☑ No central coordinator.
- ☑ Each agent would communicate with its neighbors at each iteration.
- ☑ Graphs would be used to model the communication topology.



Optimization on Networks

Model

$$\min_{\boldsymbol{x}\in\mathbb{R}^N} F(\boldsymbol{x}) = \sum_{i\in V} f_i(\boldsymbol{x})$$

- ☐ $\mathcal{G} = (V, E)$ is an undirected "decentralized" graph.
- ☐ Assume $|V| = N$ for simplicity.
- ☐ $f_i(\boldsymbol{x})$ is a local smooth and convex function.

Optimization on Networks

Model

$$\min_{\boldsymbol{x}\in\mathbb{R}^N} F(\boldsymbol{x}) = \sum_{i\in V} f_i(\boldsymbol{x})$$

- ☑ $\mathcal{G} = (V, E)$ is an undirected "decentralized" graph.
- ☐ Assume $|V| = N$ for simplicity.
- ☐ $f_i(\boldsymbol{x})$ is a local smooth and convex function.

Optimization on Networks

Model

$$\min_{\boldsymbol{x}\in\mathbb{R}^N} F(\boldsymbol{x}) = \sum_{i\in V} f_i(\boldsymbol{x})$$

- [x] $\mathcal{G} = (V, E)$ is an undirected "decentralized" graph.
- [x] Assume $|V| = N$ for simplicity.
- [ ] $f_i(\boldsymbol{x})$ is a local smooth and convex function.

Optimization on Networks

Model

$$\min_{\boldsymbol{x}\in\mathbb{R}^N} F(\boldsymbol{x}) = \textstyle\sum_{i\in V} f_i(\boldsymbol{x})$$

- ☑ $\mathcal{G} = (V, E)$ is an undirected "decentralized" graph.
- ☑ Assume $|V| = N$ for simplicity.
- ☑ $f_i(\boldsymbol{x})$ is a local smooth and convex function.

Optimization on Networks

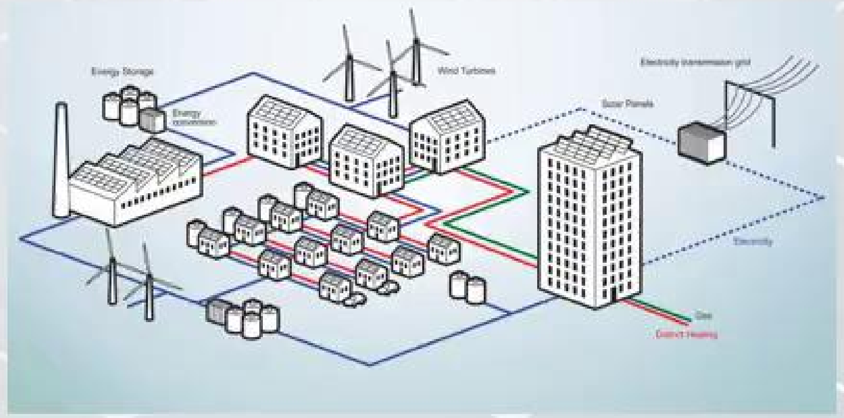
Model

$$\min_{\boldsymbol{x}\in\mathbb{R}^N} F(\boldsymbol{x}) = \sum_{i\in V} f_i(\boldsymbol{x})$$

- ☑ $\mathcal{G} = (V, E)$ is an undirected "decentralized" graph.
- ☑ Assume $|V| = N$ for simplicity.
- ☑ $f_i(\boldsymbol{x})$ is a local smooth and convex function.

Applications

multiple-agent control



distributed energy system

Consensus-based Decentralized Algorithms

$$\min_{\boldsymbol{x}\in\mathbb{R}^N} F(\boldsymbol{x}) = \sum_{i\in V} f_i(\boldsymbol{x}) \longleftrightarrow \begin{array}{l} \min\limits_{\boldsymbol{x}_i\in\mathbb{R}^N, i\in V} \quad F(\boldsymbol{x}) = \sum_{i\in V} f_i(\boldsymbol{x}_i) \\ \text{s.t.} \quad \boldsymbol{x}_i = \boldsymbol{y}, \boldsymbol{x}_j = \boldsymbol{y}, \quad (i,j)\in E \end{array}$$

Consensus-based Decentralized Algorithms

$$\min_{\boldsymbol{x}\in\mathbb{R}^N} F(\boldsymbol{x}) = \sum_{i\in V} f_i(\boldsymbol{x}) \longleftrightarrow$$

$$\begin{aligned} \min_{\boldsymbol{x}_i\in\mathbb{R}^N, i\in V} \quad & F(\boldsymbol{x}) = \sum_{i\in V} f_i(\boldsymbol{x}_i) \\ \text{s.t.} \quad & \boldsymbol{x}_i = \boldsymbol{y}, \boldsymbol{x}_j = \boldsymbol{y}, \quad (i,j) \in E \end{aligned}$$

consensus

Consensus-based Decentralized Algorithms

$$\min_{\boldsymbol{x}\in\mathbb{R}^N} F(\boldsymbol{x}) = \sum_{i\in V} f_i(\boldsymbol{x}) \longleftrightarrow \begin{array}{l} \displaystyle\min_{\boldsymbol{x}_i\in\mathbb{R}^N, i\in V} F(\boldsymbol{x}) = \sum_{i\in V} f_i(\boldsymbol{x}_i) \\ \text{s.t.} \quad \boldsymbol{x}_i = \boldsymbol{y}, \boldsymbol{x}_j = \boldsymbol{y}, \quad (i,j)\in E \end{array}$$

consensus

Consensus-based Decentralized Algorithms

$$\boldsymbol{x}_i^{k+1} = A(\boldsymbol{x}_j^k, j\in N_i) + G(\nabla f_i(\boldsymbol{x}_i^k))$$

local aggregation — gradient descent

Consensus-based Decentralized Algorithms

$$\min_{\boldsymbol{x}\in\mathbb{R}^N} F(\boldsymbol{x}) = \sum_{i\in V} f_i(\boldsymbol{x}) \longleftrightarrow \begin{array}{ll} \min\limits_{\boldsymbol{x}_i\in\mathbb{R}^N, i\in V} & F(\boldsymbol{x}) = \sum_{i\in V} f_i(\boldsymbol{x}_i) \\ \text{s.t.} & \boldsymbol{x}_i = \boldsymbol{y}, \boldsymbol{x}_j = \boldsymbol{y}, \quad (i,j)\in E \end{array}$$

consensus

Consensus-based Decentralized Algorithms

$$\boldsymbol{x}_i^{k+1} = A(\boldsymbol{x}_j^k, j\in N_i) + G(\nabla f_i(\boldsymbol{x}_i^k))$$

local aggregation gradient descent

- Examples: DGD, D-ADMM, EXACT, PG-EXTRA, NIDS...
- Distributed: each agent deals with local computation.
- Decentralized: each agent only communicates with its neighbors.

$$\min_{\boldsymbol{x}\in\mathbb{R}^N} F(\boldsymbol{x}) = \sum_{i\in V} f_i(\boldsymbol{x}) \longleftrightarrow \begin{array}{ll} \displaystyle\min_{\boldsymbol{x}_i\in\mathbb{R}^N, i\in V} & F(\boldsymbol{x}) = \sum_{i\in V} f_i(\boldsymbol{x}_i) \\ \text{s.t.} & \boldsymbol{x}_i = \boldsymbol{y}, \boldsymbol{x}_j = \boldsymbol{y}, \quad (i,j)\in E \end{array}$$

consensus

Consensus-based Decentralized Algorithms

$$\boldsymbol{x}_i^{k+1} = A(\boldsymbol{x}_j^k, j \in N_i) + G(\nabla f_i(\boldsymbol{x}_i^k))$$

local aggregation   gradient descent

- ✓ Examples: DGD, D-ADMM, EXACT, PG-EXTRA, NIDS...
- ○ Distributed: each agent deals with local computation.
- ○ Decentralized: each agent only communicates with its neighbors.

Consensus-based Decentralized Algorithms

$$\min_{\boldsymbol{x}\in\mathbb{R}^N} F(\boldsymbol{x}) = \sum_{i\in V} f_i(\boldsymbol{x}) \longleftrightarrow \begin{array}{ll} \min\limits_{\boldsymbol{x}_i\in\mathbb{R}^N, i\in V} & F(\boldsymbol{x}) = \sum_{i\in V} f_i(\boldsymbol{x}_i) \\ \text{s.t.} & \boldsymbol{x}_i = \boldsymbol{y}, \boldsymbol{x}_j = \boldsymbol{y}, \quad (i,j)\in E \end{array}$$

consensus

Consensus-based Decentralized Algorithms

$$\boldsymbol{x}_i^{k+1} = A(\boldsymbol{x}_j^k, j\in N_i) + G(\nabla f_i(\boldsymbol{x}_i^k))$$

local aggregation gradient descent

- ☑ Examples: DGD, D-ADMM, EXACT, PG-EXTRA, NIDS...
- ☑ Distributed: each agent deals with local computation.
- ○ Decentralized: each agent only communicates with its neighbors.

Consensus-based Decentralized Algorithms

$$\min_{\boldsymbol{x}\in\mathbb{R}^N} F(\boldsymbol{x}) = \sum_{i\in V} f_i(\boldsymbol{x}) \longleftrightarrow \begin{array}{ll} \displaystyle\min_{\boldsymbol{x}_i\in\mathbb{R}^N, i\in V} & F(\boldsymbol{x}) = \sum_{i\in V} f_i(\boldsymbol{x}_i) \\ \text{s.t.} & \boldsymbol{x}_i = \boldsymbol{y}, \boldsymbol{x}_j = \boldsymbol{y}, \quad (i,j)\in E \end{array}$$

consensus

Consensus-based Decentralized Algorithms

$$\boldsymbol{x}_i^{k+1} = A(\boldsymbol{x}_j^k, j\in N_i) + G(\nabla f_i(\boldsymbol{x}_i^k))$$

local aggregation gradient descent

- Examples: DGD, D-ADMM, EXACT, PG-EXTRA, NIDS...
- Distributed: each agent deals with local computation.
- Decentralized: each agent only communicates with its neighbors.

Consensus-based Decentralized Algorithms

$$\min_{\boldsymbol{x}\in\mathbb{R}^N} F(\boldsymbol{x}) = \sum_{i\in V} f_i(\boldsymbol{x}) \longleftrightarrow \begin{array}{l} \min\limits_{\boldsymbol{x}_i\in\mathbb{R}^N, i\in V} F(\boldsymbol{x}) = \sum_{i\in V} f_i(\boldsymbol{x}_i) \\ \text{s.t.} \quad \boldsymbol{x}_i = \boldsymbol{y}, \boldsymbol{x}_j = \boldsymbol{y}, \quad (i,j)\in E \end{array}$$

consensus

Consensus-based Decentralized Algorithms

$$\boldsymbol{x}_i^{k+1} = A(\boldsymbol{x}_j^k, j\in N_i) + G(\nabla f_i(\boldsymbol{x}_i^k))$$

local aggregation — gradient descent

- Examples: DGD, D-ADMM, EXACT, PG-EXTRA, NIDS...
- Distributed: each agent deals with local computation.
- Decentralized: each agent only communicates with its neighbors.

Question

Could we also distribute the global variable $\boldsymbol{x}$ into local blocks?

Spatially Distributed Systems

Spatially Distributed Systems

- Example: $\min_{\boldsymbol{x}} \|\mathsf{A}\boldsymbol{x} - \boldsymbol{b}\|^2 = \sum_i (\mathsf{A}_i^T \boldsymbol{x} - \boldsymbol{b}_i)^2$
- Assumption: A has off-diagonal decay.
  - polynomial decay: $|A_{ij}| \leq C(1 + |i-j|)^{-\alpha}, \quad \alpha > 1$
  - exponential decay: $|A_{ij}| \leq Ce^{-\gamma|i-j|}, \quad \gamma > 0$
- Wiener's Lemma: A^{-1} has a similar off-diagonal decay as A.
  - the component $x_i = (\mathsf{A}^{-1}\boldsymbol{b}))_i$ depends mostly on the neighbors of b_i.

Spatially Distributed Systems

- ☑ Example: $\min_{\boldsymbol{x}} \|\mathsf{A}\boldsymbol{x} - \boldsymbol{b}\|^2 = \sum_i (\mathsf{A}_i^T \boldsymbol{x} - \boldsymbol{b}_i)^2$
- ☐ Assumption: A has off-diagonal decay.
  - polynomial decay: $|A_{ij}| \leq C(1 + |i - j|)^{-\alpha}, \quad \alpha > 1$
  - exponential decay: $|A_{ij}| \leq Ce^{-\gamma|i-j|}, \quad \gamma > 0$
- ☐ Wiener's Lemma: A^{-1} has a similar off-diagonal decay as A.
  - the component $x_i = (\mathsf{A}^{-1}\boldsymbol{b})_i$ depends mostly on the neighbors of b_i.

Spatially Distributed Systems

- ☑ Example: $\min_{\boldsymbol{x}} \|\mathsf{A}\boldsymbol{x} - \boldsymbol{b}\|^2 = \sum_i (\mathsf{A}_i^T \boldsymbol{x} - \boldsymbol{b}_i)^2$
- ☑ Assumption: A has off-diagonal decay.
  - polynomial decay: $|A_{ij}| \leq C(1 + |i - j|)^{-\alpha}, \quad \alpha > 1$
  - exponential decay: $|A_{ij}| \leq Ce^{-\gamma|i-j|}, \quad \gamma > 0$
- ☐ Wiener's Lemma: A^{-1} has a similar off-diagonal decay as A.
  - the component $x_i = (\mathsf{A}^{-1}\boldsymbol{b}))_i$ depends mostly on the neighbors of b_i.

Spatially Distributed Systems

- [x] Example: $\min_{\boldsymbol{x}} \|\mathsf{A}\boldsymbol{x} - \boldsymbol{b}\|^2 = \sum_i (\mathsf{A}_i^T \boldsymbol{x} - \boldsymbol{b}_i)^2$
- [x] Assumption: A has off-diagonal decay.
  - (✓) polynomial decay: $|A_{ij}| \leq C(1 + |i - j|)^{-\alpha}, \quad \alpha > 1$
  - ○ exponential decay: $|A_{ij}| \leq Ce^{-\gamma|i-j|}, \quad \gamma > 0$
- [ ] Wiener's Lemma: A^{-1} has a similar off-diagonal decay as A.
  - ○ the component $x_i = (\mathsf{A}^{-1}\boldsymbol{b})_i$ depends mostly on the neighbors of b_i.

Spatially Distributed Systems

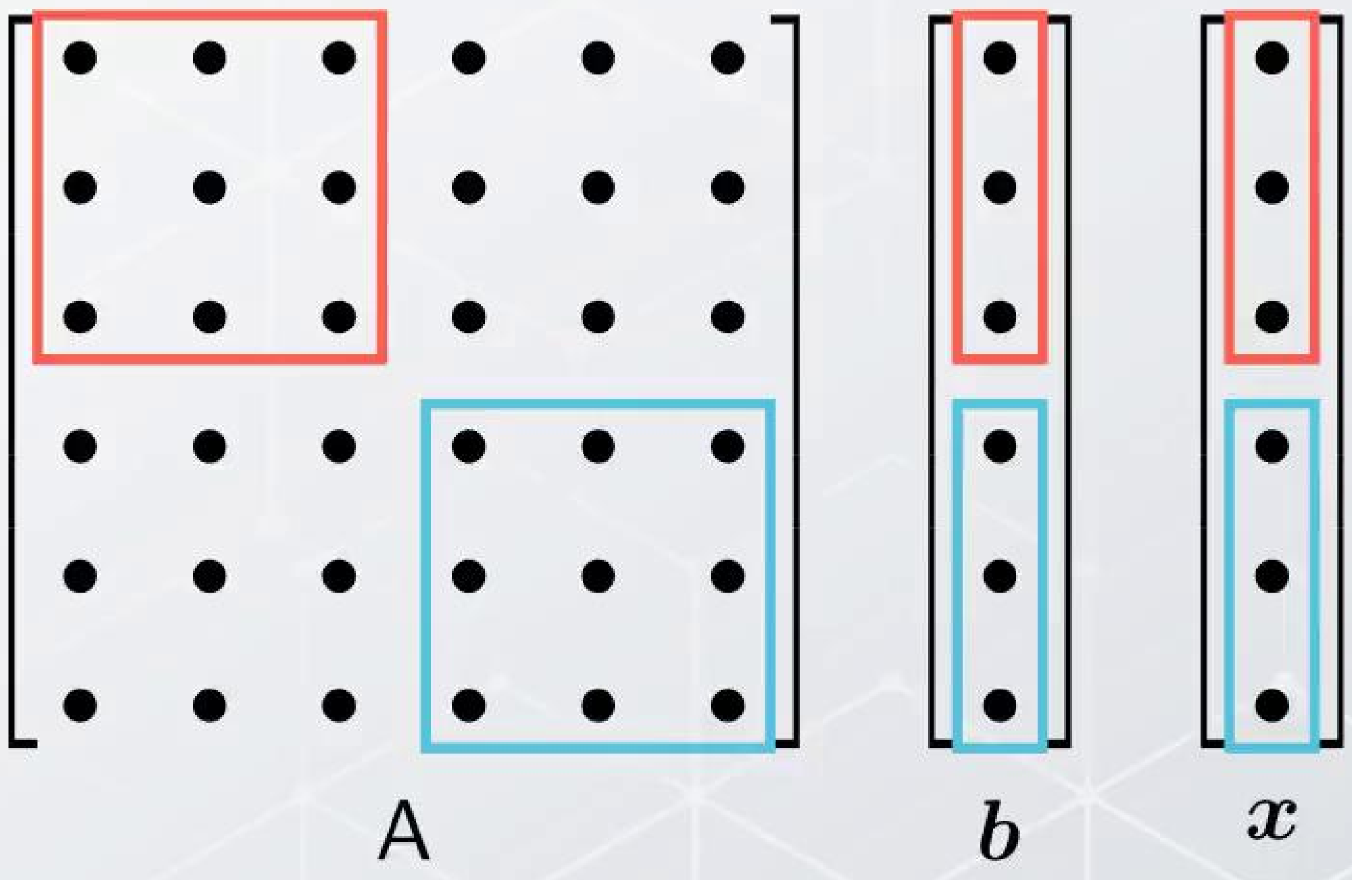
- ☑ Example: $\min_{\boldsymbol{x}} \|\mathsf{A}\boldsymbol{x} - \boldsymbol{b}\|^2 = \sum_i (\mathsf{A}_i^T \boldsymbol{x} - \boldsymbol{b}_i)^2$
- ☑ Assumption: A has off-diagonal decay.
  - polynomial decay: $|A_{ij}| \leq C(1 + |i - j|)^{-\alpha}, \quad \alpha > 1$
  - exponential decay: $|A_{ij}| \leq Ce^{-\gamma|i-j|}, \quad \gamma > 0$
- ☐ Wiener's Lemma: A^{-1} has a similar off-diagonal decay as A.
  - the component $x_i = (\mathsf{A}^{-1}\boldsymbol{b}))_i$ depends mostly on the neighbors of b_i.

Spatially Distributed Systems

- Example: $\min_{\boldsymbol{x}} \|\mathsf{A}\boldsymbol{x} - \boldsymbol{b}\|^2 = \sum_i (\mathsf{A}_i^T \boldsymbol{x} - \boldsymbol{b}_i)^2$
- Assumption: A has off-diagonal decay.
  - polynomial decay: $|A_{ij}| \leq C(1 + |i - j|)^{-\alpha}, \quad \alpha > 1$
  - exponential decay: $|A_{ij}| \leq Ce^{-\gamma|i-j|}, \quad \gamma > 0$
- Wiener's Lemma: A^{-1} has a similar off-diagonal decay as A.
  - the component $x_i = (\mathsf{A}^{-1}\boldsymbol{b})_i$ depends mostly on the neighbors of b_i.

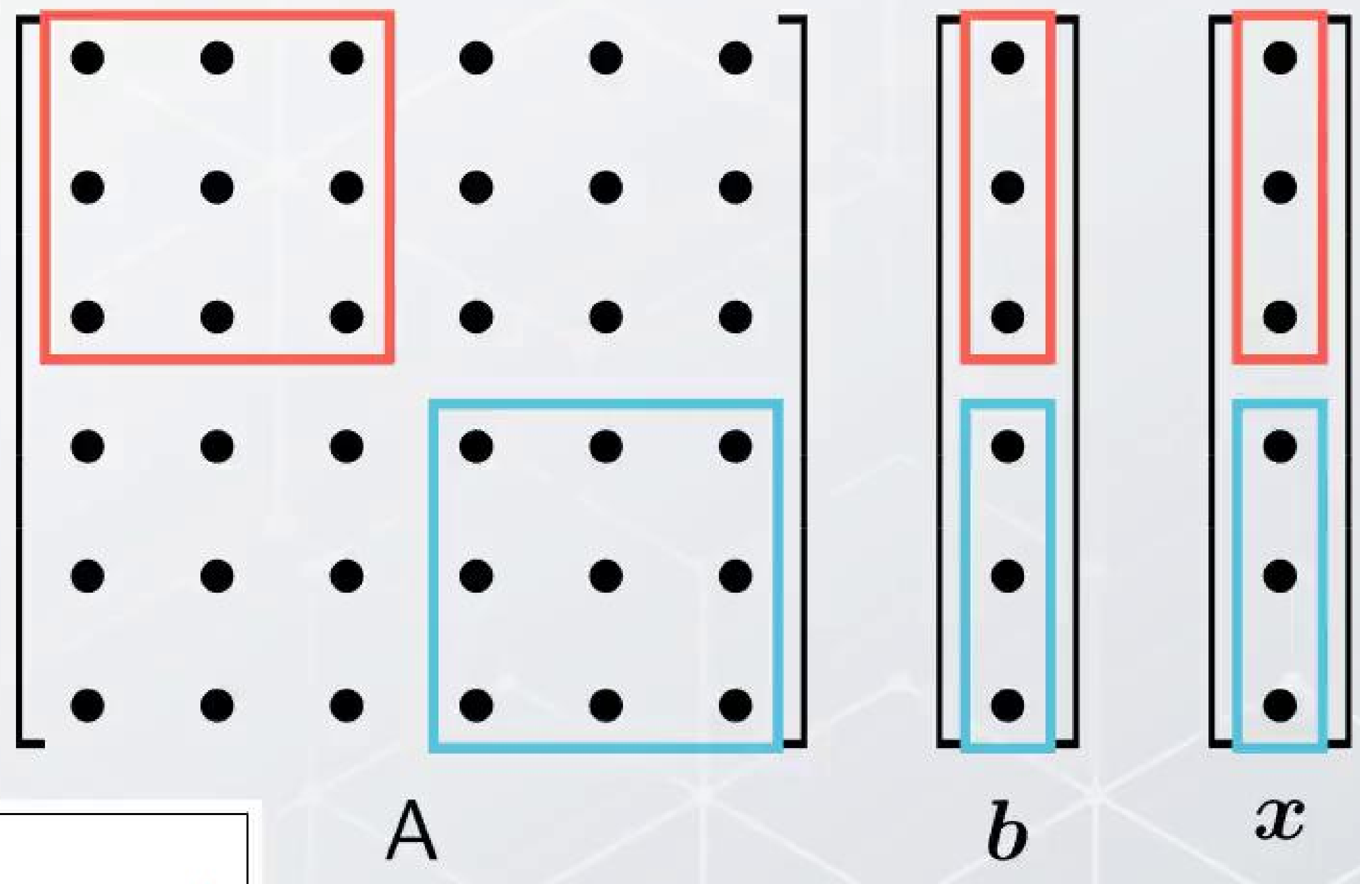
Spatially Distributed Systems

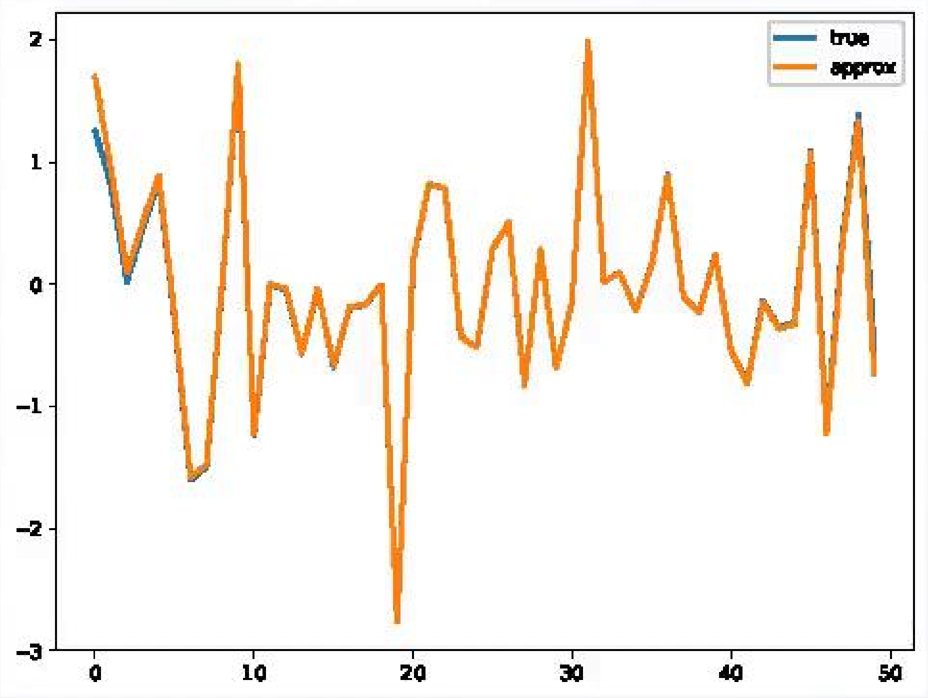
- ☑ Example: $\min_{\boldsymbol{x}} \|\mathsf{A}\boldsymbol{x} - \boldsymbol{b}\|^2 = \sum_i (\mathsf{A}_i^T \boldsymbol{x} - \boldsymbol{b}_i)^2$
- ☑ Assumption: A has off-diagonal decay.
  - polynomial decay: $|A_{ij}| \leq C(1 + |i - j|)^{-\alpha}, \quad \alpha > 1$
  - exponential decay: $|A_{ij}| \leq C e^{-\gamma|i-j|}, \quad \gamma > 0$
- ☑ Wiener's Lemma: A^{-1} has a similar off-diagonal decay as A.
  - the component $x_i = (\mathsf{A}^{-1}\boldsymbol{b})_i$ depends mostly on the neighbors of b_i.



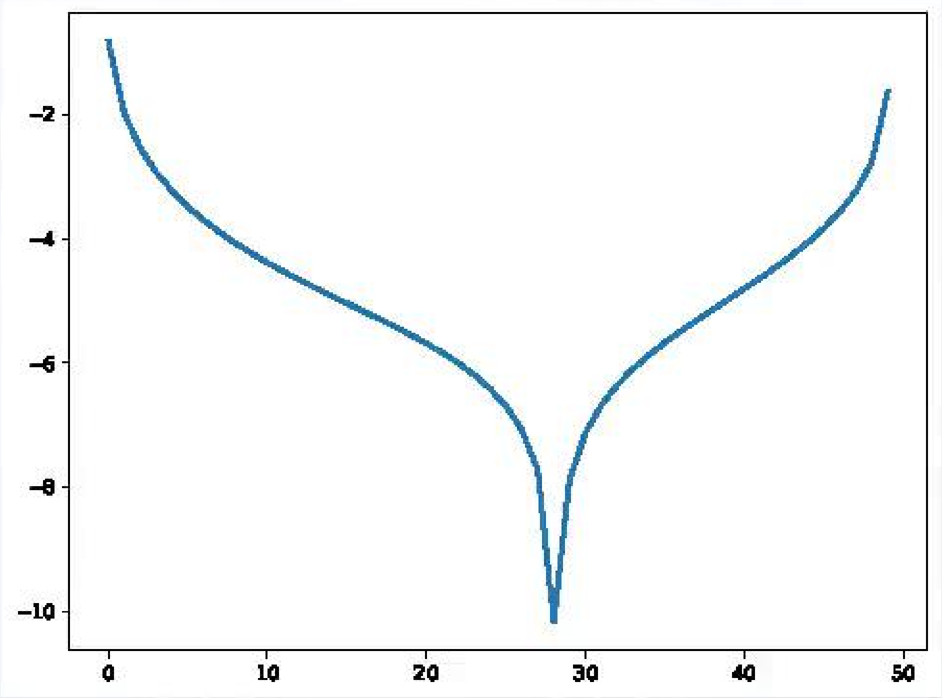
Spatially Distributed Systems

- Example: $\min_{\boldsymbol{x}} \|\mathsf{A}\boldsymbol{x} - \boldsymbol{b}\|^2 = \sum_i (\mathsf{A}_i^T \boldsymbol{x} - \boldsymbol{b}_i)^2$
- Assumption: A has off-diagonal decay.
  - polynomial decay: $|A_{ij}| \leq C(1 + |i-j|)^{-\alpha}, \quad \alpha > 1$
  - exponential decay: $|A_{ij}| \leq C e^{-\gamma|i-j|}, \quad \gamma > 0$
- Wiener's Lemma: A^{-1} has a similar off-diagonal decay as A.
  - the component $x_i = (\mathsf{A}^{-1}\boldsymbol{b})_i$ depends mostly on the neighbors of b_i.





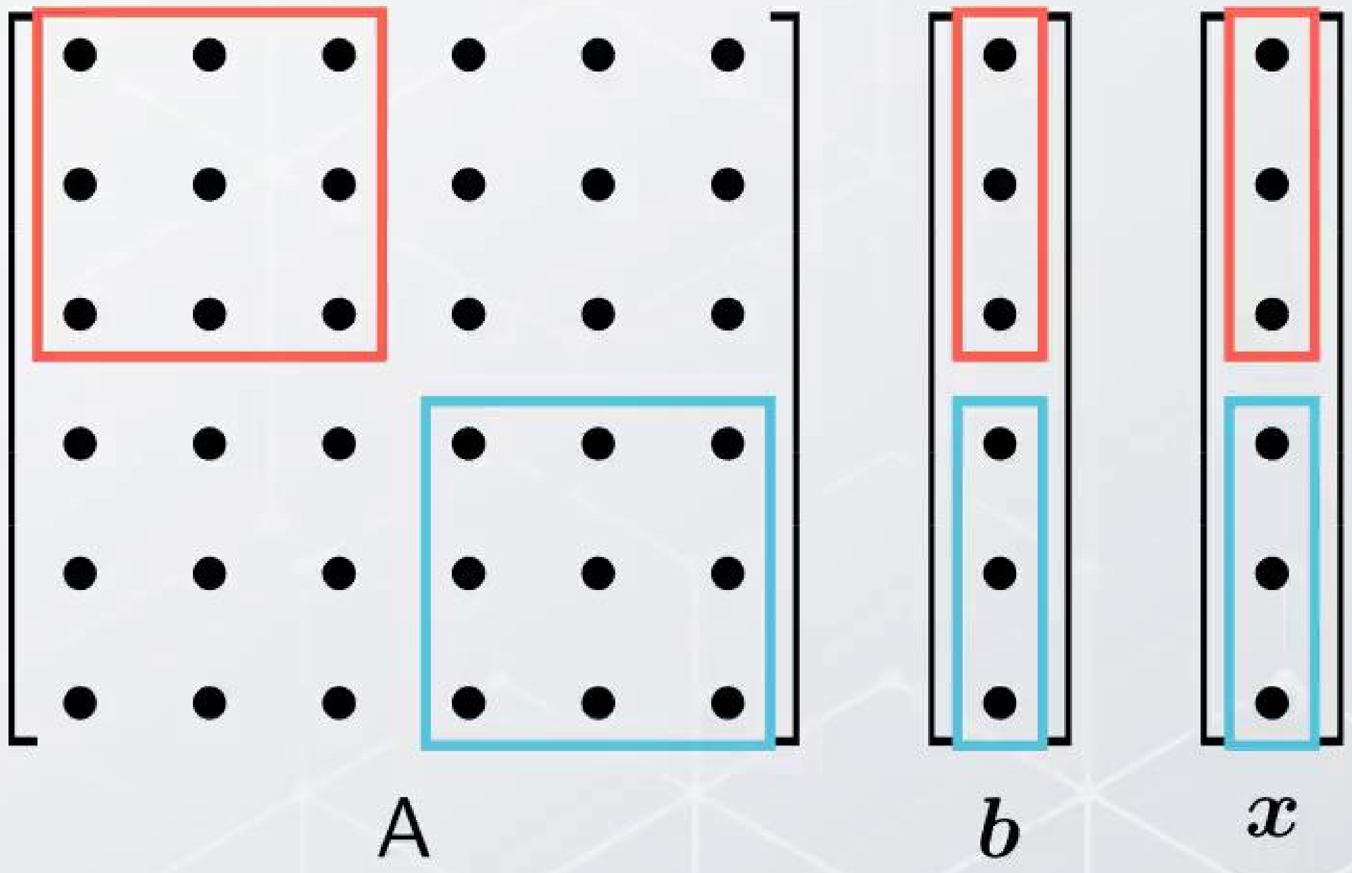
true vs. approximation.



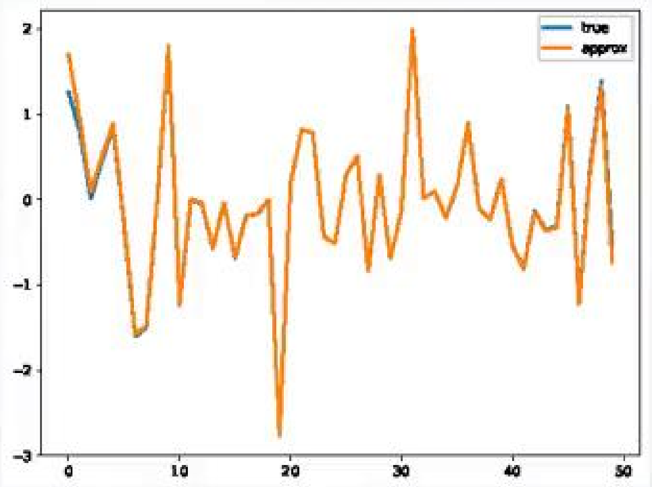
component-wise log error.

Spatially Distributed Systems

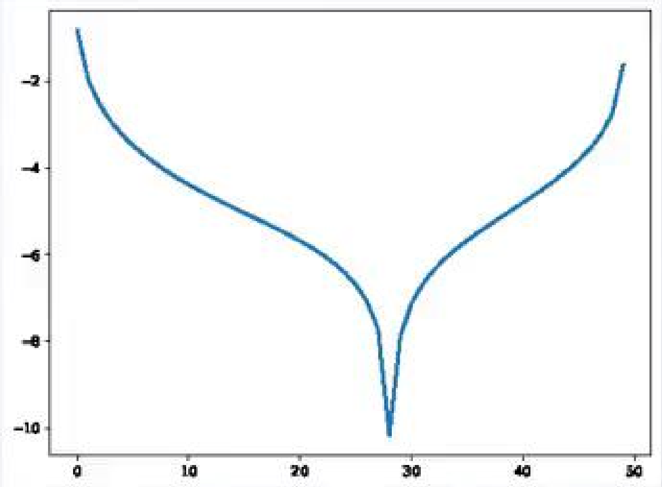
- ☑ Example: $\min_{\boldsymbol{x}} \|\mathsf{A}\boldsymbol{x} - \boldsymbol{b}\|^2 = \sum_i (\mathsf{A}_i^T \boldsymbol{x} - \boldsymbol{b}_i)^2$
- ☑ Assumption: A has off-diagonal decay.
  - polynomial decay: $|A_{ij}| \le C(1 + |i - j|)^{-\alpha}, \quad \alpha > 1$
  - exponential decay: $|A_{ij}| \le Ce^{-\gamma|i-j|}, \quad \gamma > 0$
- ☑ Wiener's Lemma: A^{-1} has a similar off-diagonal decay as A.
  - the component $x_i = (\mathsf{A}^{-1}\boldsymbol{b})_i$ depends mostly on the neighbors of b_i.



Use overlapped blocks of $\boldsymbol{x}$.



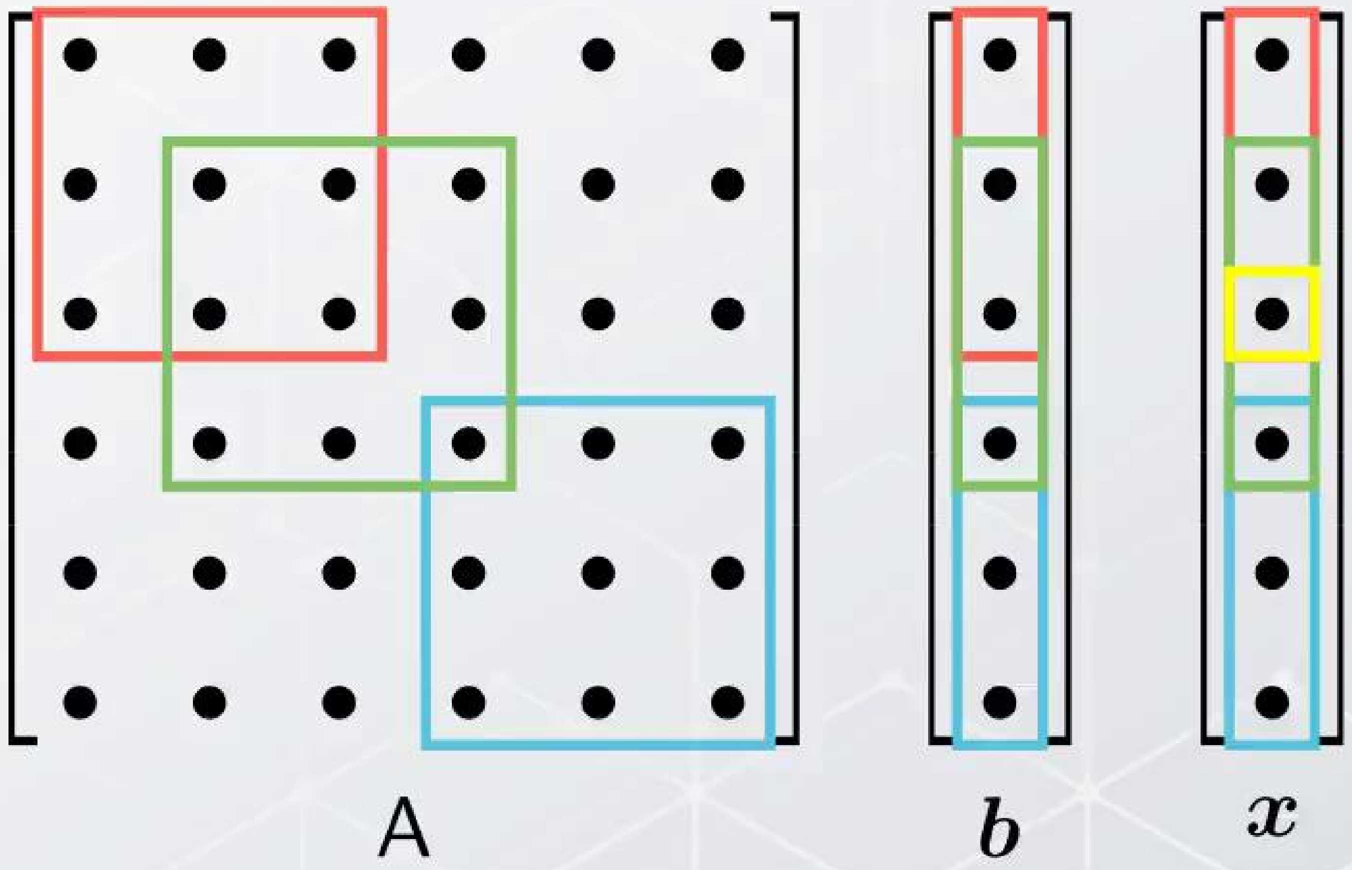
true vs. approximation.



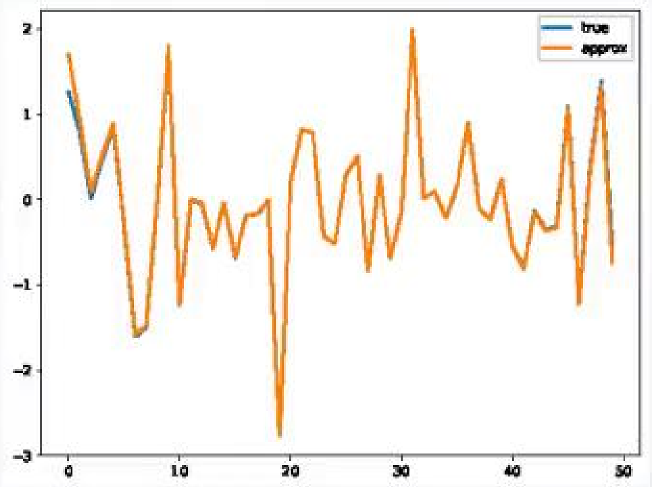
component-wise log error.

Spatially Distributed Systems

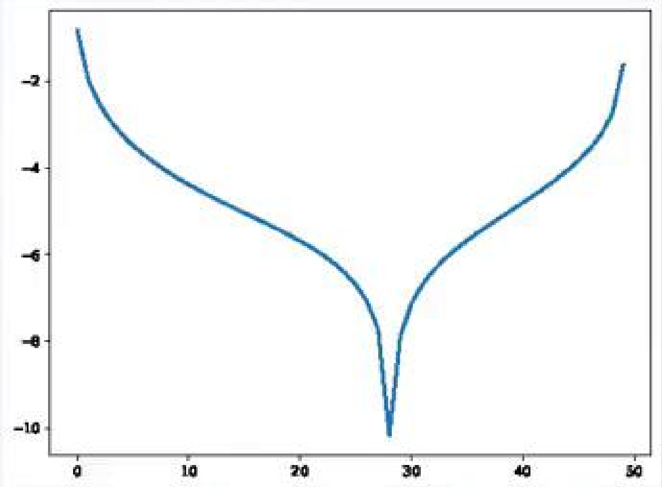
- Example: $\min_{\boldsymbol{x}} \|\mathsf{A}\boldsymbol{x} - \boldsymbol{b}\|^2 = \sum_i (\mathsf{A}_i^T \boldsymbol{x} - \boldsymbol{b}_i)^2$
- Assumption: A has off-diagonal decay.
  - polynomial decay: $|A_{ij}| \le C(1 + |i-j|)^{-\alpha}, \quad \alpha > 1$
  - exponential decay: $|A_{ij}| \le Ce^{-\gamma|i-j|}, \quad \gamma > 0$
- Wiener's Lemma: A^{-1} has a similar off-diagonal decay as A.
  - the component $x_i = (\mathsf{A}^{-1}\boldsymbol{b})_i$ depends mostly on the neighbors of b_i.



Use overlapped blocks of $\boldsymbol{x}$.



true vs. approximation.



component-wise log error.

Spatially Distributed Network

Spatially Distributed Network

- $\mathcal{G} = (V, E)$ is an undirected "decentralized" graph.
- The counting measure μ has polynomial growth:

  $$\mu(B(i,r)) \le C(1+r)^d, \quad i \in V, r > 0$$

- V could be divided into a family of domains.
  - Partition: $V = \bigcup_{\lambda \in \Lambda} D_\lambda$, where λ is a fusion center.
  - There exists R-neighbors $D_{\lambda,R}$ of D_λ such that $D_\lambda \subseteq D_{\lambda,R}$ and $\rho(D_\lambda, V \backslash D_{\lambda,R}) > R$.
- Each local function f_i is banded: it only depends on x_j for $j \in B(i,m)$.

Spatially Distributed Network

- ☑ $\mathcal{G} = (V, E)$ is an undirected "decentralized" graph.
- ☐ The counting measure μ has polynomial growth:

$$\mu(B(i,r)) \leq C(1+r)^d, \quad i \in V, r > 0$$

- ☐ V could be divided into a family of domains.
  - Partition: $V = \bigcup_{\lambda \in \Lambda} D_\lambda$, where λ is a fusion center.
  - There exists R-neighbors $D_{\lambda,R}$ of D_λ such that $D_\lambda \subseteq D_{\lambda,R}$ and $\rho(D_\lambda, V \backslash D_{\lambda,R}) > R$.
- ☐ Each local function f_i is banded: it only depends on x_j for $j \in B(i,m)$.

Spatially Distributed Network

- ☑ $\mathcal{G} = (V, E)$ is an undirected “decentralized” graph.
- ☑ The counting measure μ has polynomial growth:

$$\mu(B(i,r)) \le C(1+r)^d, \quad i \in V, r > 0$$

- ☐ V could be divided into a family of domains.
  - Partition: $V = \bigcup_{\lambda\in\Lambda} D_\lambda$, where λ is a fusion center.
  - There exists R-neighbors $D_{\lambda,R}$ of D_λ such that $D_\lambda \subseteq D_{\lambda,R}$ and $\rho(D_\lambda, V\backslash D_{\lambda,R}) > R$.
- ☐ Each local function f_i is banded: it only depends on x_j for $j \in B(i,m)$.

Spatially Distributed Network

- ☑ $\mathcal{G} = (V, E)$ is an undirected "decentralized" graph.
- ☑ The counting measure μ has polynomial growth:

  $$\mu(B(i,r)) \le C(1+r)^d, \quad i \in V, r > 0$$

- ☑ V could be divided into a family of domains.
  - Partition: $V = \bigcup_{\lambda \in \Lambda} D_\lambda$, where λ is a fusion center.
  - There exists R-neighbors $D_{\lambda,R}$ of D_λ such that $D_\lambda \subseteq D_{\lambda,R}$ and $\rho(D_\lambda, V \backslash D_{\lambda,R}) > R$.
- ☐ Each local function f_i is banded: it only depends on x_j for $j \in B(i,m)$.

Spatially Distributed Network

- [x] $\mathcal{G} = (V, E)$ is an undirected "decentralized" graph.
- [x] The counting measure μ has polynomial growth:

  $$\mu(B(i,r)) \leq C(1+r)^d, \quad i \in V, r > 0$$
- [x] V could be divided into a family of domains.
  - ⊙ Partition: $V = \bigcup_{\lambda \in \Lambda} D_\lambda$, where λ is a fusion center.
  - ○ There exists R-neighbors $D_{\lambda,R}$ of D_λ such that $D_\lambda \subseteq D_{\lambda,R}$ and $\rho(D_\lambda, V \backslash D_{\lambda,R}) > R$.
- [ ] Each local function f_i is banded: it only depends on x_j for $j \in B(i,m)$.

Spatially Distributed Network

- ☑ $\mathcal{G} = (V, E)$ is an undirected "decentralized" graph.
- ☑ The counting measure μ has polynomial growth:

  $$\mu(B(i,r)) \le C(1+r)^d, \quad i \in V, r > 0$$

- ☑ V could be divided into a family of domains.
  - Partition: $V = \bigcup_{\lambda \in \Lambda} D_\lambda$, where λ is a fusion center.
  - There exists R-neighbors $D_{\lambda,R}$ of D_λ such that $D_\lambda \subseteq D_{\lambda,R}$ and $\rho(D_\lambda, V \backslash D_{\lambda,R}) > R$.
- ☐ Each local function f_i is banded: it only depends on x_j for $j \in B(i,m)$.

Spatially Distributed Network

- $\mathcal{G} = (V, E)$ is an undirected "decentralized" graph.
- The counting measure μ has polynomial growth:

  $$\mu(B(i,r)) \leq C(1+r)^d, \quad i \in V, r > 0$$
- V could be divided into a family of domains.
  - Partition: $V = \bigcup_{\lambda \in \Lambda} D_\lambda$, where λ is a fusion center.
  - There exists R-neighbors $D_{\lambda,R}$ of D_λ such that $D_\lambda \subseteq D_{\lambda,R}$ and $\rho(D_\lambda, V \backslash D_{\lambda,R}) > R$.
- Each local function f_i is banded: it only depends on x_j for $j \in B(i,m)$.

Decentralized Algorithm

Decentralized Algorithm

- Iterative Distributed/Decentralized Algorithm
  - local minimization: $\boldsymbol{w}_\lambda^{(n)} = \mathrm{argmin}_{\boldsymbol{u}} F(\chi^*_{D_{\lambda,R}} \boldsymbol{u} + I_{V\setminus D_{\lambda,R}} \boldsymbol{x}^{(n)})$
  - weighted aggregation: $\boldsymbol{x}^{(n+1)} = \sum_{\lambda\in\Lambda} I_{D_\lambda} \chi^*_{D_{\lambda,R}} \boldsymbol{w}_\lambda^{(n)}$

Decentralized Algorithm

- Iterative Distributed/Decentralized Algorithm
  - local minimization: $\boldsymbol{w}_\lambda^{(n)} = \mathrm{argmin}_{\boldsymbol{u}} F(\chi^*_{D_{\lambda,R}} \boldsymbol{u} + I_{V \setminus D_{\lambda,R}} \boldsymbol{x}^{(n)})$
  - weighted aggregation: $\boldsymbol{x}^{(n+1)} = \sum_{\lambda \in \Lambda} I_{D_\lambda} \chi^*_{D_{\lambda,R}} \boldsymbol{w}_\lambda^{(n)}$

Decentralized Algorithm

- Iterative Distributed/Decentralized Algorithm
  - local minimization: $\boldsymbol{w}_\lambda^{(n)} = \mathrm{argmin}_{\boldsymbol{u}} F(\chi^*_{D_{\lambda,R}}\boldsymbol{u} + I_{V\setminus D_{\lambda,R}}\boldsymbol{x}^{(n)})$
  - weighted aggregation: $\boldsymbol{x}^{(n+1)} = \sum_{\lambda\in\Lambda} I_{D_\lambda}\chi^*_{D_{\lambda,R}}\boldsymbol{w}_\lambda^{(n)}$

Decentralized Algorithm

- Iterative Distributed/Decentralized Algorithm
  - local minimization: $\boldsymbol{w}_\lambda^{(n)} = \mathrm{argmin}_{\boldsymbol{u}} F(\chi_{D_{\lambda,R}}^* \boldsymbol{u} + I_{V \setminus D_{\lambda,R}} \boldsymbol{x}^{(n)})$
  - weighted aggregation: $\boldsymbol{x}^{(n+1)} = \sum_{\lambda \in \Lambda} I_{D_\lambda} \chi_{D_{\lambda,R}}^* \boldsymbol{w}_\lambda^{(n)}$

Decentralized Algorithm

- Iterative Distributed/Decentralized Algorithm
  - local minimization: $\boldsymbol{w}_\lambda^{(n)} = \mathrm{argmin}_{\boldsymbol{u}} F(\chi^*_{D_{\lambda,R}}\boldsymbol{u} + I_{V\setminus D_{\lambda,R}}\boldsymbol{x}^{(n)})$
  - weighted aggregation: $\boldsymbol{x}^{(n+1)} = \sum_{\lambda\in\Lambda} I_{D_\lambda}\chi^*_{D_{\lambda,R}}\boldsymbol{w}_\lambda^{(n)}$

Convergence Theorem ([Emirov, S., and Sun, 2024])

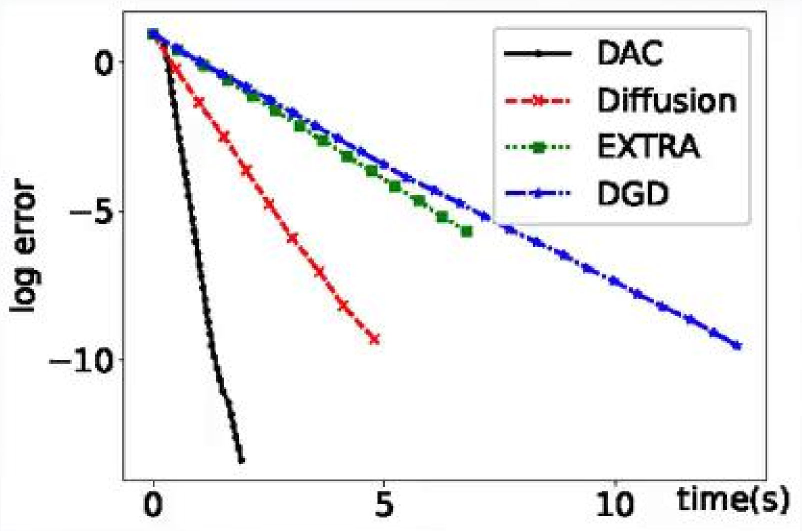
$$\|\boldsymbol{x}^{(n)} - \boldsymbol{x}^*\|_p \le C(\delta_R)^n\|\boldsymbol{x}^{(0)} - \boldsymbol{x}^*\|_p, \quad 1 \le p \le \infty$$

where $\delta_R = c(1-\beta/L)^{(R-2m-1)/(2m)}(R+1)^d$

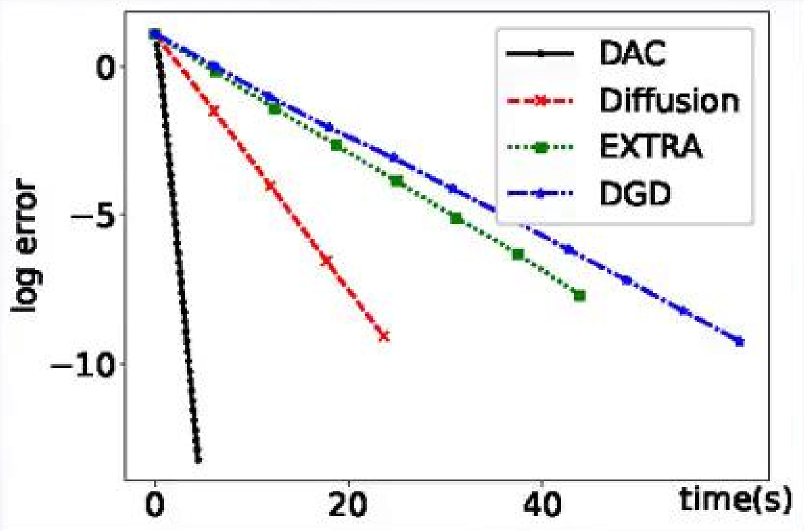
Emirov, Song, and Sun, A Divide-and-Conquer Algorithm for Distributed Optimization on Networks, Applied and Computational Harmonic Analysis, 70(2024)

Numerical Experiments: Least Squares

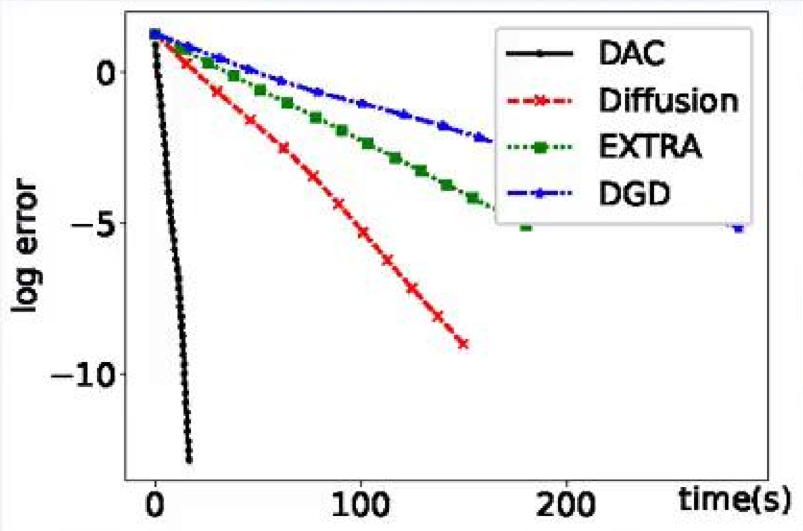
- Consider $F(\boldsymbol{x}) = \frac{1}{2}\|\mathsf{H}\boldsymbol{x} - \boldsymbol{b}\|_2^2$
  - $\mathsf{H} = \mathrm{I} + 5\mathsf{L}_{\mathcal{G}}$, where $\mathsf{L}_{\mathcal{G}}$ is the graph Laplacian.
  - $\boldsymbol{b}$ is randomly chosen.



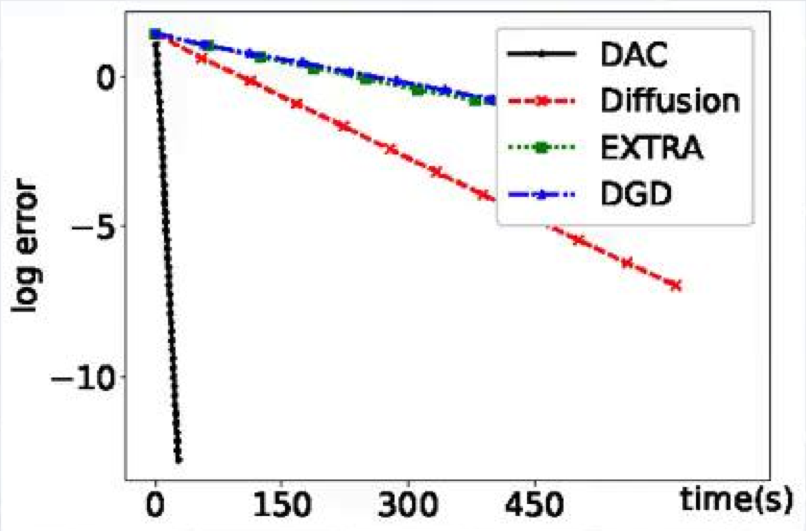
$N = 256$



$N = 512$



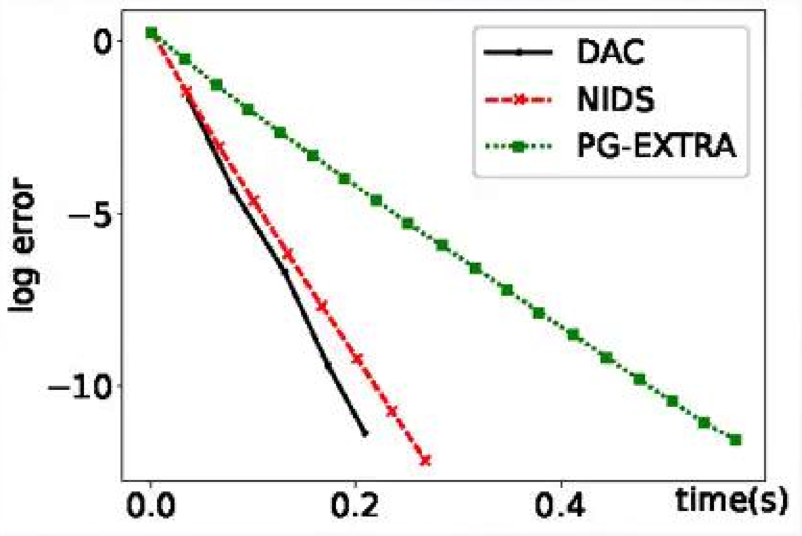
$N = 1024$



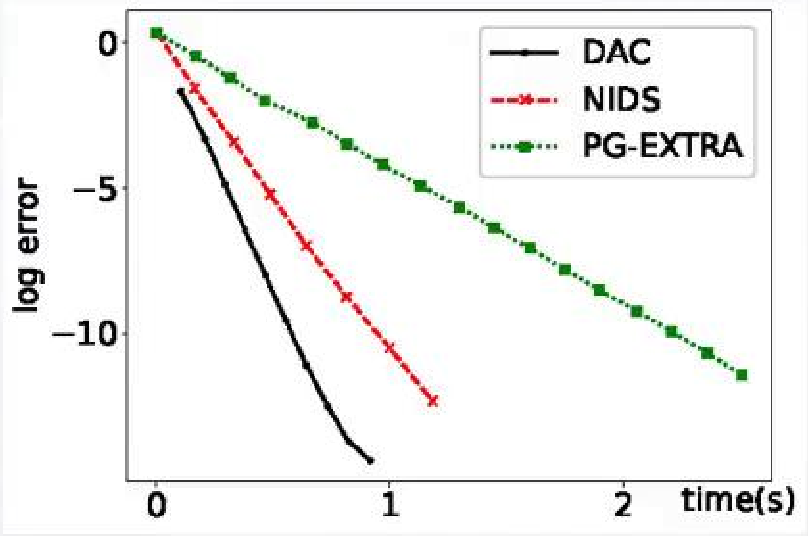
$N = 2048$

Numerical Experiments: LASSO

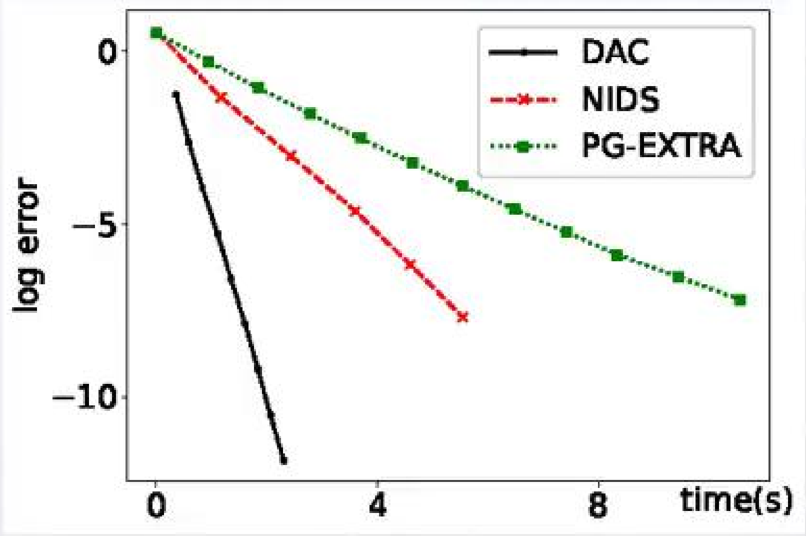
- Consider $F(\boldsymbol{x}) = \frac{1}{2}\|\mathsf{H}\boldsymbol{x} - \boldsymbol{b}\|_2^2 + \mu\|\boldsymbol{x}\|_1$
  - $\mathsf{H} = \mathrm{I} + 5\mathsf{L}_{\mathcal{G}}$, where $\mathsf{L}_{\mathcal{G}}$ is the graph Laplacian.
  - $\boldsymbol{b}$ is randomly chosen.



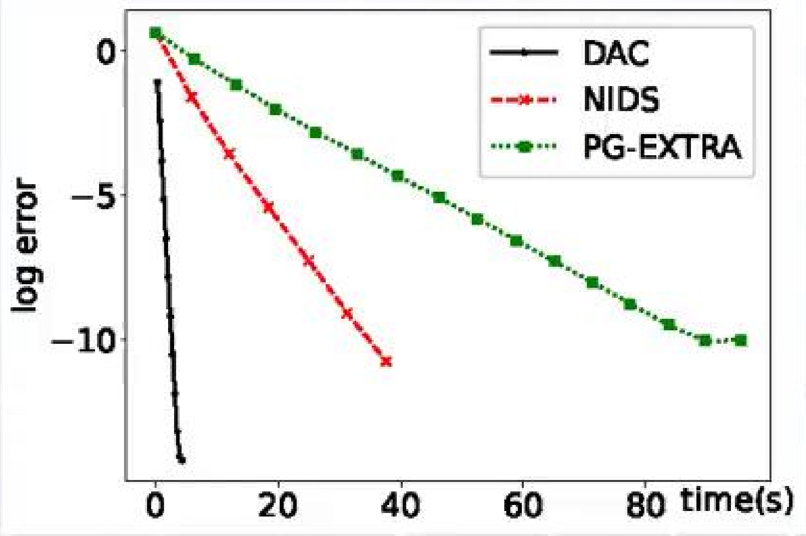
$N = 256$



$N = 512$



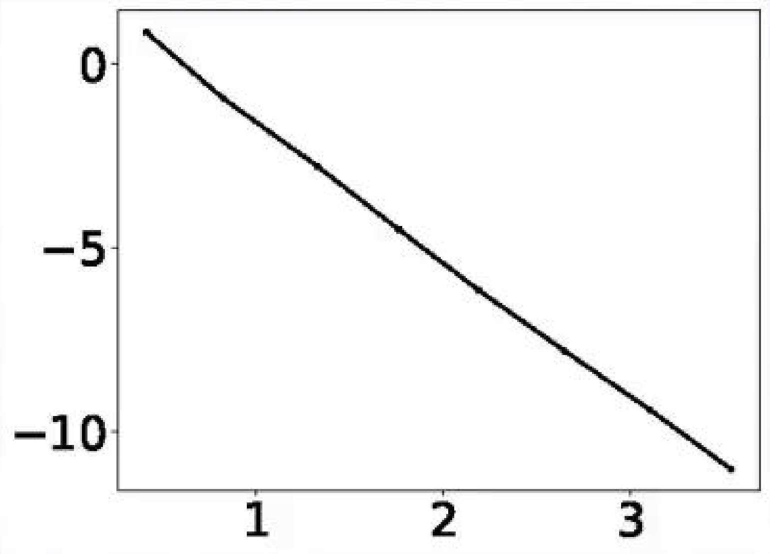
$N = 1024$



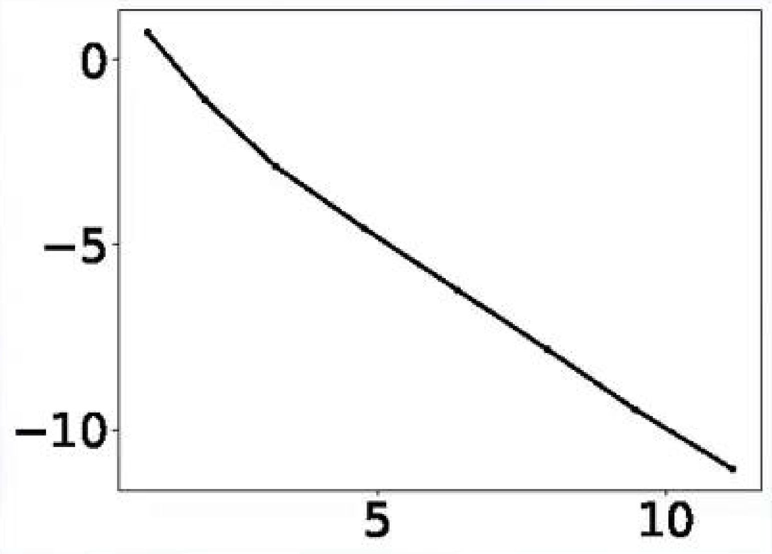
$N = 2048$

Numerical Experiments: SVM

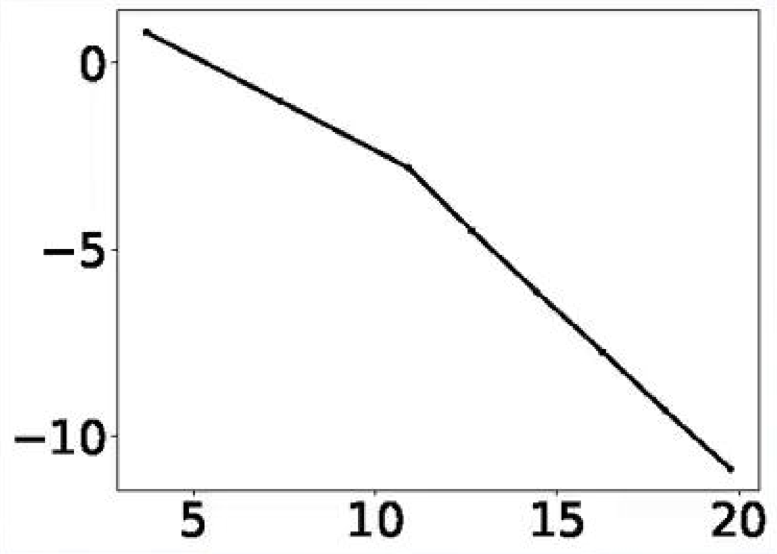
- Consider $F(\boldsymbol{x}) = \sum_i \max\left\{0, 1 - y_i \sum_j H_{i,j} x_j\right\} + \mu \|\boldsymbol{x}\|_1$
  - $\mathsf{H} = \mathrm{I} + 5\mathsf{L}_{\mathcal{G}}$, where $\mathsf{L}_{\mathcal{G}}$ is the graph Laplacian.
  - $\boldsymbol{y}$ is randomly chosen.



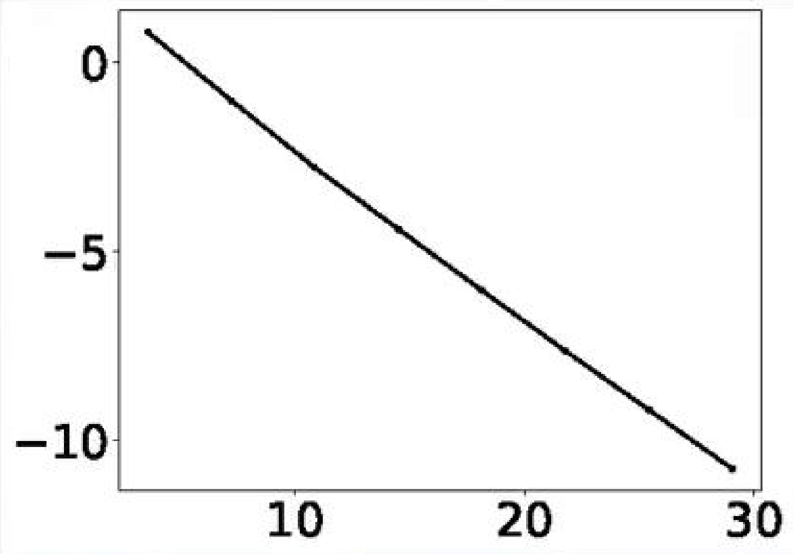
$N = 256$



$N = 512$



$N = 1024$



$N = 2048$

Outlook

- Summary
  - The proposed method is distributed in both the objective functions and the variables.
  - It relies on the spatially local structures of the problem.
  - It works well for both smooth and non-smooth problems.
- References

  - Emirov, Song, and Sun, A Divide-and-Conquer Algorithm for Distributed Optimization on Networks, Applied and Computational Harmonic Analysis, 70(2024)

- Contact: Guohui Song

  - Email: gsong@odu.edu
  - Webpage: gsong-math.github.io

Outlook

- Summary
  - The proposed method is distributed in both the objective functions and the variables.
  - It relies on the spatially local structures of the problem.
  - It works well for both smooth and non-smooth problems.
- References
  - Emirov, Song, and Sun, A Divide-and-Conquer Algorithm for Distributed Optimization on Networks, Applied and Computational Harmonic Analysis, 70(2024)
- Contact: Guohui Song
  - Email: gsong@odu.edu
  - Webpage: gsong-math.github.io