ADMM and Fast Gradient Methods for Distributed Optimization

João Xavier
Instituto Sistemas e Robótica (ISR), Instituto Superior Técnico (IST)

European Control Conference, ECC'13
July 16, 2013

Joint work

Fast gradient methods

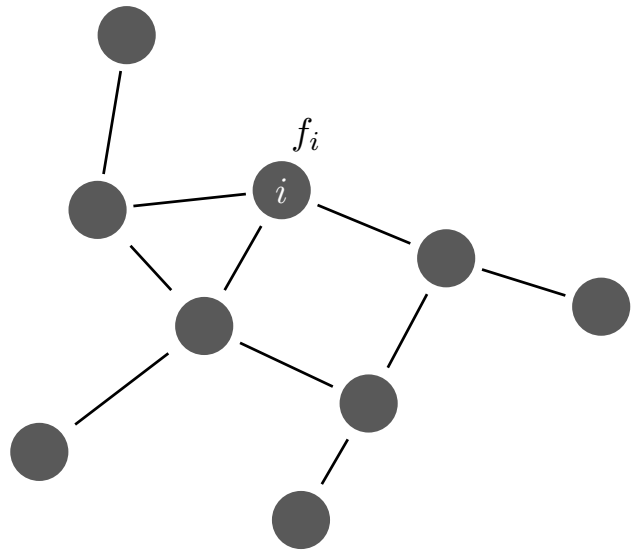
- Dušan Jakovetić (IST-CMU)
- José M. F. Moura (CMU)

ADMM

- João Mota (IST-CMU)
- Markus Püschel (ETH)
- Pedro Aguiar (IST)

Multi-agent optimization

$$\begin{array}{ll} \text{minimize} & f(x) := f_1(x) + f_2(x) + \cdots + f_n(x) \\ \text{subject to} & x \in \mathcal{X} \end{array}$$



- f_i is convex, private to agent i
- $\mathcal{X} \subset \mathbf{R}^d$ is closed, convex (hereafter, $d = 1$)
- $f^\star = \inf_{x \in \mathcal{X}} f(x)$ is attained at $x^\star$
- network is connected and static
- applications: distributed learning, cognitive radio, consensus, ...

Distributed subgradient method

Update at each agent i (with constant step size)

$$x_i(t) = \mathcal{P}_{\mathcal{X}} \left(\sum_{j \in \mathcal{N}_i} W_{ij}\, x_j(t-1) - \alpha \nabla f_i(x_i(t-1)) \right)$$

- $\mathcal{N}_i$ is neighborhood of node i (including i)
- W_{ij} are weights
- $\alpha > 0$ is stepsize
- $\nabla f_i(x)$ is a subgradient of f_i at x
- $\mathcal{P}_{\mathcal{X}}$ is projector onto $\mathcal{X}$

Several variations exist and time-varying networks are supported

Small sample of representative work:

- J .Tsitsiklis *et al.*, "Distributed asynchronous deterministic and stochastic gradient optimization algorithms," *IEEE TAC*, 31(9), 1986
- A. Nedić and A. Ozdaglar, "Distributed subgradient methods for multi-agent optimization," *IEEE TAC*, 54(1), 2009
- B. Johansson, "A randomized incremental subgradient method for distributed optimization in networked systems," *SIAM J. Optimization*, 20(3), 2009
- A. Nedić *et al.*, "Constrained consensus and optimization in multi-agent networks," *IEEE TAC*, 55(4), 2010
- J. Duchi *et al.*, "Dual averaging for distributed optimization: convergence and network scaling," *IEEE TAC*, 57(3), 2012

Convergence analysis: under appropriate conditions

$$f(x_i(t)) - f^\star = \mathcal{O}\left(\alpha + \frac{1}{\alpha t}\right)$$

For optimized α, $\mathcal{O}(1/\epsilon^2)$ iterations suffice to reach ϵ-suboptimality

Distributed subgradient method in matrix form

$$x(t) = \mathcal{P}\left(W\, x(t-1) - \alpha \nabla F(x(t-1))\right)$$

- $x(t) := (x_1(t), \ldots, x_n(t))$ is network state
- $F(x_1, \ldots, x_n) := f_1(x_1) + \cdots + f_n(x_n)$
- $\nabla F(x_1, \ldots, x_n) = (\nabla f_1(x_1), \ldots, \nabla f_n(x_n))$
- $\mathcal{P}$ is projector onto $\mathcal{X}^n$

Interpretation:

- when $W = I - \alpha\rho\mathcal{L}$ ($\mathcal{L}$ = network Laplacian, $\rho > 0$)

$$x(t) = \mathcal{P}\left(x(t-1) - \alpha\nabla\Psi_\rho(x(t-1))\right)$$

- classical subgradient method applied to penalized objective

$$\Psi_\rho(x) = F(x) + \frac{\rho}{2}x^\top\mathcal{L}x = \sum_{i=1}^{n} f_i(x_i) + \frac{\rho}{2}\sum_{i\sim j}\|x_i - x_j\|^2$$

Key idea: apply instead Nesterov's fast gradient method

$$\begin{aligned} x(t) &= \mathcal{P}\left(y(t-1) - \alpha\nabla\Psi_\rho(y(t-1))\right) \\ y(t) &= x(t) + \frac{t-1}{t+2}\left(x(t) - x(t-1)\right) \end{aligned}$$

- $y(t) = (y_1(t), \ldots, y_n(t))$ is auxiliary variable

Distributed Nesterov gradient method (D-NG) with constant stepsize

$$\begin{aligned} x(t) &= \mathcal{P}\left(W\, y(t-1) - \alpha \nabla F(y(t-1))\right) \\ y(t) &= x(t) + \frac{t-1}{t+2}\left(x(t) - x(t-1)\right) \end{aligned}$$

Convergence analysis: if f_i's are differentiable, ∇f_i's are Lipschitz continuous, and $\mathcal{X}$ is compact

$$f(x_i(t)) - f^\star = \mathcal{O}\left(\frac{1}{\rho} + \frac{1}{t^2} + \frac{\rho}{t^2}\right)$$

For optimized ρ, $\mathcal{O}(1/\epsilon)$ iterations suffice to reach ϵ-suboptimality

D. Jakovetić *et al.*, "Distributed Nesterov-like gradient algorithms", *IEEE* 51*st Annual Conference on Decision and Control (CDC)*, 2012

Proof

Step 1: plug-in Nesterov's classical analysis

- $\|\nabla f_i(x) - \nabla f_i(y)\| \leq L \|x - y\|$ implies

$$\|\nabla \Psi_\rho(x) - \nabla \Psi_\rho(y)\| \leq L_\rho \|x - y\|$$

for $L_\rho = L + \rho \lambda_{\max}(\mathcal{L})$

- notation: $\Psi_\rho^\star := \inf_{x \in \mathcal{X}} \Psi_\rho(x)$ is attained at $x_\rho^\star$

- with $\alpha := 1/L_\rho$, classical analysis yields

$$\begin{aligned} \Psi_\rho(x(t)) - \Psi_\rho^\star &\leq \frac{2L_\rho}{t^2} \left\|x(0) - x_\rho^\star\right\|^2 \\ &\leq \frac{2L_\rho}{t^2} B \end{aligned} \tag{1}$$

for some $B \geq 0$ (since $\mathcal{X}$ is compact)

Step 2: relate $f(x_i)$ to $\Psi_\rho(x)$, $x = (x_1, \ldots, x_n)$

$$\begin{aligned} f(x_i) &= \sum_{j=1}^{n} f_j(x_i) \\ &= \underbrace{\sum_{j=1}^{n} f_j(x_j) + \frac{\rho}{2} x^\top \mathcal{L} x}_{\Psi_\rho(x)} + \underbrace{\sum_{j=1}^{n} f_j(x_i) - f_j(x_j) - \frac{\rho}{2} x^\top \mathcal{L} x}_{\Delta(x)} \end{aligned} \tag{2}$$

Step 3: upper bound $\Delta(x) \leq \frac{C}{\rho}$ for some $C \geq 0$

- use Lipschitz continuity of f_j to obtain

$$\sum_{j=1}^{n} f_j(x_i) - f_j(x_j) \leq G \sum_{j=1}^{n} \|x_i - x_j\| \tag{3}$$

for some $G \geq 0$

- since $\mathcal{L}\mathbf{1} = 0$,

$$x^\top \mathcal{L} x = (x - x_i \mathbf{1})^\top \mathcal{L} (x - x_i \mathbf{1}) \tag{4}$$

- combine (3) and (4) to obtain

$$\begin{aligned} \Delta(x) &\leq G \|x - x_i \mathbf{1}\|_1 - \frac{\rho}{2} (x - x_i \mathbf{1})^\top \mathcal{L} (x - x_i \mathbf{1}) \\ &= G \|\widehat{x}\|_1 - \frac{\rho}{2} \widehat{x}^\top \widehat{\mathcal{L}} \widehat{x} \end{aligned}$$

  - $\widehat{x}$ is $x - x_i \mathbf{1}$ with ith entry removed
  - $\widehat{\mathcal{L}}$ is $\mathcal{L}$ with ith row and *ith* column removed
  - easy to see that $\widehat{\mathcal{L}}$ is positive definite (network is connected)

- it follows

$$\begin{aligned}
\Delta(x) &\leq \max_y G\,\|y\|_1 - \frac{\rho}{2} y^\top \widehat{\mathcal{L}} y \\
&= \max_y \max_{\|z\|_\infty \leq 1} G z^\top y - \frac{\rho}{2} y^\top \widehat{\mathcal{L}} y \\
&= \max_{\|z\|_\infty \leq 1} \max_y G z^\top y - \frac{\rho}{2} y^\top \widehat{\mathcal{L}} y \\
&= \frac{1}{\rho} \underbrace{\frac{G^2}{2} \max_{\|z\|_\infty \leq 1} z^\top \widehat{\mathcal{L}}^{-1} z}_{C} \qquad (5)
\end{aligned}$$

- use $f^\star \geq \Psi_\rho^\star$, and combine (1), (2) with (5) to conclude

$$f(x_i(t)) - f^\star \leq \frac{2(L + \rho \lambda_{\max}(\mathcal{L}))B}{t^2} + \frac{C}{\rho} = \mathcal{O}\left(\frac{1}{\rho} + \frac{1}{t^2} + \frac{\rho}{t^2}\right) \blacksquare$$

Numerical example

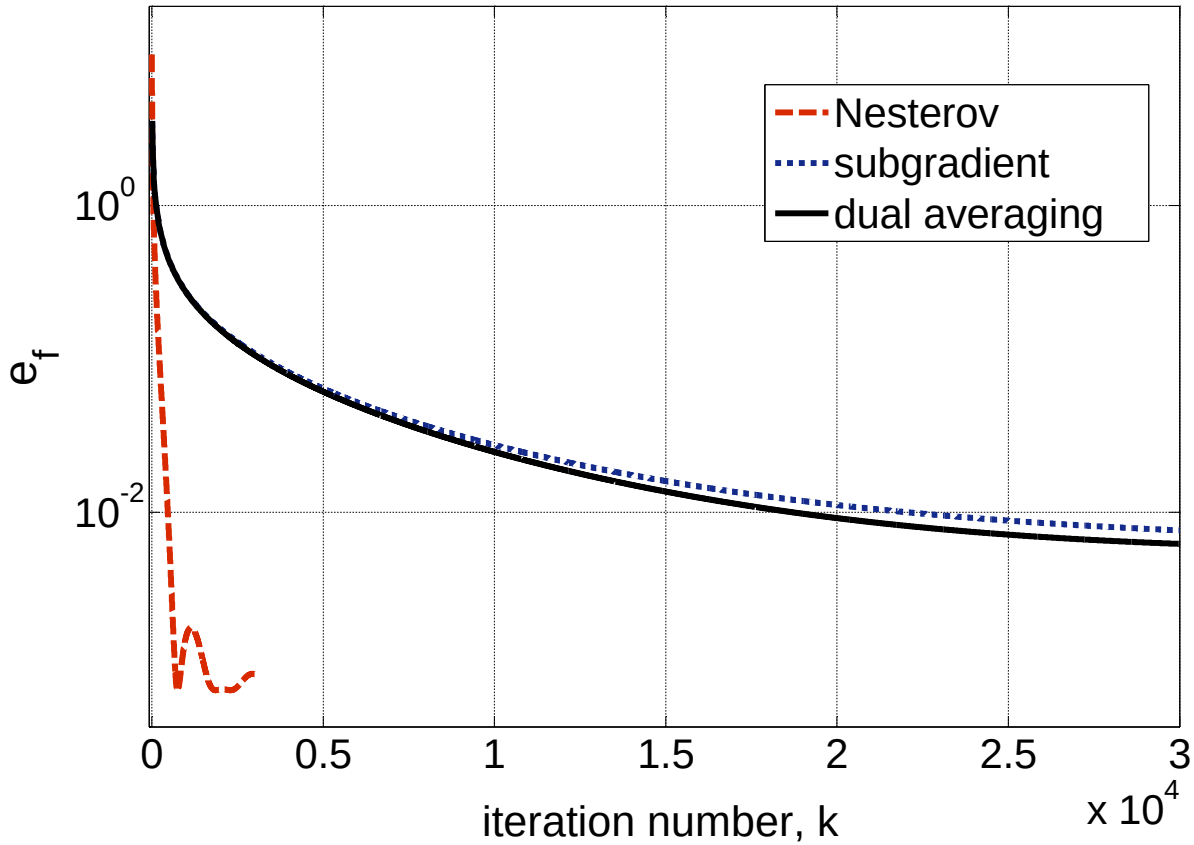
Distributed logistic regression

$$\begin{array}{ll} \underset{s,r}{\text{minimize}} & \sum_{i=1}^{n} \underbrace{\sum_{j=1}^{5} \phi\left(-b_{ij}(s^\top a_{ij} + r)\right)}_{f_i(s,r)} \\ \text{subject to} & \|s\| \leq R \end{array}$$

- $\{(a_{ij}, b_{ij}) \in \mathbb{R}^3 \times \mathbb{R} : j = 1, \ldots, 5\}$: training data for agent i
- $\phi(t) = \log\left(1 + e^{-t}\right)$
- geometric graph, $n = 20$ nodes and 86 edges

Constant stepsize

$$e_f(t) := \frac{1}{n}\sum_{i=1}^{n} \frac{f(x_i(t)) - f^\star}{f^\star}$$



Red: D-NG[1] Blue: subgradient[2] Black: dual averaging[3]

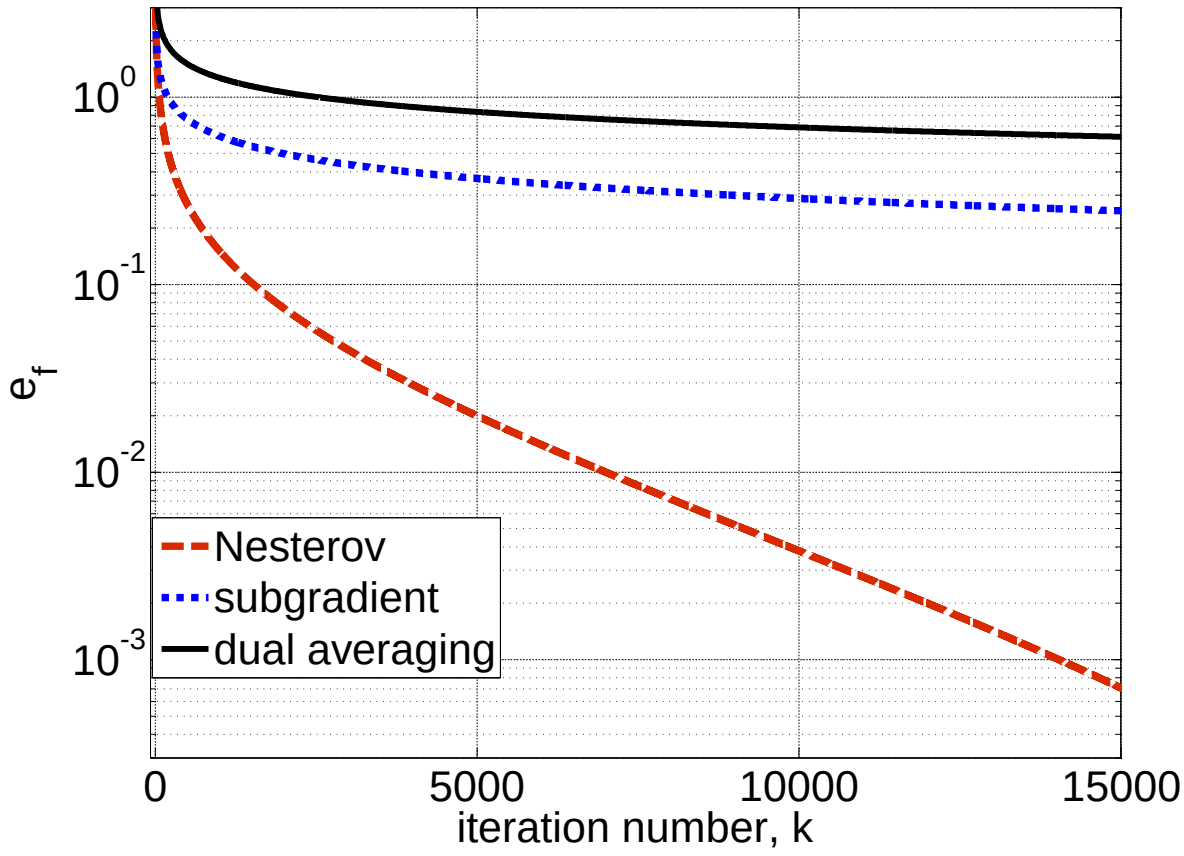
---

[1] D. Jakovetić *et al.*, "Distributed Nesterov-like gradient algorithms", *IEEE* 51*st Annual Conference on Decision and Control (CDC)*, 2012

[2] A. Nedić and A. Ozdaglar, "Distributed subgradient methods for multi-agent optimization," *IEEE TAC*, 54(1), 2009

[3] J. Duchi *et al.*, "Dual averaging for distributed optimization: convergence and network scaling," *IEEE TAC*, 57(3), 2012

Diminishing stepsize $e_f(t) := \frac{1}{n}\sum_{i=1}^{n} \frac{f(x_i(t))-f^\star}{f^\star}$



D-NG: $\alpha(t) = 1/t$ Subgradient, dual averaging: $\alpha(t) = 1/\sqrt{t}$

Distributed Nesterov gradient method (D-NG) with diminishing stepsize

Unconstrained problem

$$\begin{array}{ll} \text{minimize} & f(x) := f_1(x) + f_2(x) + \cdots + f_n(x) \\ \text{subject to} & x \in \mathbf{R}^d \end{array}$$

Diminishing stepsize $\alpha(t) = \frac{c}{t+1}$

$$x(t) = W\, y(t-1) - \alpha(t-1)\nabla F(y(t-1)) \quad (6)$$

$$y(t) = x(t) + \frac{t-1}{t+2}\,(x(t) - x(t-1)) \quad (7)$$

D. Jakovetić *et al.*, "Fast cooperative distributed learning", 46*th Asilomar Conference on Signals, Systems and Computers (ASILOMAR)*, 2012

Convergence analysis: if f_i's are differentiable, ∇f_i's are bounded and L-Lipschitz continuous, and W is symmetric, stochastic and pos.-def.

$$f(x_i(t)) - f^\star = \mathcal{O}\left(\frac{\log t}{t}\right)$$

- dependence on network spectral gap $\frac{1}{1-\lambda_2(W)}$ also known
- agents may ignore L and $\lambda_2(W)$

Sample of related work (different assumptions):

- K. Tsianos and M. Rabbat, "Distributed strongly convex optimization," 50*th Allerton Conference on Communication, Control and Computing*, 2012
- E. Ghadimi *et al.*, "Accelerated gradient methods for networked optimization," arXiv preprint, 2012

Sketch of proof

Step 1: look at network averages

$$\overline{x}(t) := \frac{1}{n}\sum_{i=1}^{n} x_i(t) \qquad \overline{y}(t) := \frac{1}{n}\sum_{i=1}^{n} y_i(t)$$

- from (6) and (7)

$$\begin{aligned}
\overline{x}(t) &= \overline{y}(t-1) - \underbrace{\frac{\alpha(t-1)}{n}}_{\frac{1}{\widehat{L}(t-1)}} \underbrace{\sum_{i=1}^{n} \nabla f_i(y_i(t-1))}_{\widehat{\nabla} f(\overline{y}(t-1))} \\
\overline{y}(t) &= \overline{x}(t) + \frac{t-1}{t+2}\left(\overline{x}(t) - \overline{x}(t-1)\right)
\end{aligned}$$

- interpretation: inexact Nesterov's gradient method

- using ideas from optimization with inexact oracles[4],

$$f(\overline{x}(t)) - f^\star \leq \frac{2n}{ct}\left\|\overline{x}(0) - x^\star\right\|^2 + \frac{L}{t}\sum_{s=0}^{t-1}\frac{(s+2)^2}{s+1}\left\|\delta_y(s)\right\|^2 \qquad (8)$$

where

$$\begin{aligned}\delta_y(t) &:= y(t) - \overline{y}(t)\mathbf{1} \\ &= (y_1(t) - \overline{y}(t), \ldots, y_n(t) - \overline{y}(t))\end{aligned}$$

[4] O. Devolder *et al.*, "First-order methods of smooth convex optimization with inexact oracle," submitted, *Mathematical Programming*, 2011

Step 2: show $\delta_y(t) = \mathcal{O}\left(\frac{1}{t}\right)$

- rewrite (6) and (7) as the time-varying linear system

$$\begin{bmatrix} \delta_x(t) \\ \delta_x(t-1) \end{bmatrix} = A(t) \begin{bmatrix} \delta_x(t-1) \\ \delta_x(t-2) \end{bmatrix} + u(t-1), \tag{9}$$

  where

$$A(t) := \begin{bmatrix} \frac{2t-1}{t+1}\Delta_W & -\frac{t-2}{t+1}\Delta_W \\ I & 0 \end{bmatrix}, \quad u(t) := \begin{bmatrix} -\alpha(t)(I-J)\nabla F(y(t)) \\ 0 \end{bmatrix}$$

  - $\delta_x(t) := x(t) - \overline{x}(t)\mathbf{1}$
  - $J := \frac{1}{n}\mathbf{1}\mathbf{1}^\top$
  - $\Delta_W := W - J$

- $u(t) = \mathcal{O}\left(\frac{1}{t}\right)$ due to $\alpha(t) = \frac{c}{t+1}$ and bounded gradient assumption

- Fact: if

$$x(t) = \lambda x(t-1) + \mathcal{O}\left(\frac{1}{t}\right) \tag{10}$$

with $|\lambda| < 1$, then $x(t) = \mathcal{O}\left(\frac{1}{t}\right)$

- "hand-waving" argument: upon approximating

$$A(t) \simeq \begin{bmatrix} 2\Delta_W & -\Delta_W \\ I & 0 \end{bmatrix},$$

and diagonalizing, system (9) reduces to (10)

- since $\delta_x(t) = \mathcal{O}\left(\frac{1}{t}\right)$,

$$\begin{aligned} \delta_y(t) &= \delta_x(t) + \frac{t-1}{t+2}\left(\delta_x(t) - \delta_x(t-1)\right) \\ &= \mathcal{O}\left(\frac{1}{t}\right) \end{aligned}$$

Step 3: relate $f(x_i)$ to $f(\overline{x})$

$$\begin{aligned} f(x_i) &= \sum_{j=1}^{n} f_j(x_i) \\ &= \underbrace{\sum_{j=1}^{n} f_j(\overline{x})}_{f(\overline{x})} + \underbrace{\sum_{j=1}^{n} f_j(x_i) - f_j(\overline{x})}_{\Delta(x)} \end{aligned} \tag{11}$$

- by the bounded gradient assumption

$$\Delta(x) = \sum_{j=1}^{n} f_j(x_i) - f_j(\overline{x}) \leq Gn \left\| x_i - \overline{x} \right\| \leq Gn \left\| \delta_x \right\| \tag{12}$$

- combine (8), (11) and (12) to conclude

$$f(x_i(t)) - f^\star = \mathcal{O}\left(\frac{\log t}{t}\right) \blacksquare$$

Numerical example

Acoustic source localization in sensor networks

- agent i measures

$$y_i = \frac{1}{\|x - r_i\|^2} + \text{noise}$$

  r_i = position of agent i

- goal: determine source position x
- convex approach[5]:

$$\underset{x}{\text{minimize}} \quad \sum_{i=1}^{n} \text{dist}^2\left(x, C_i\right)$$

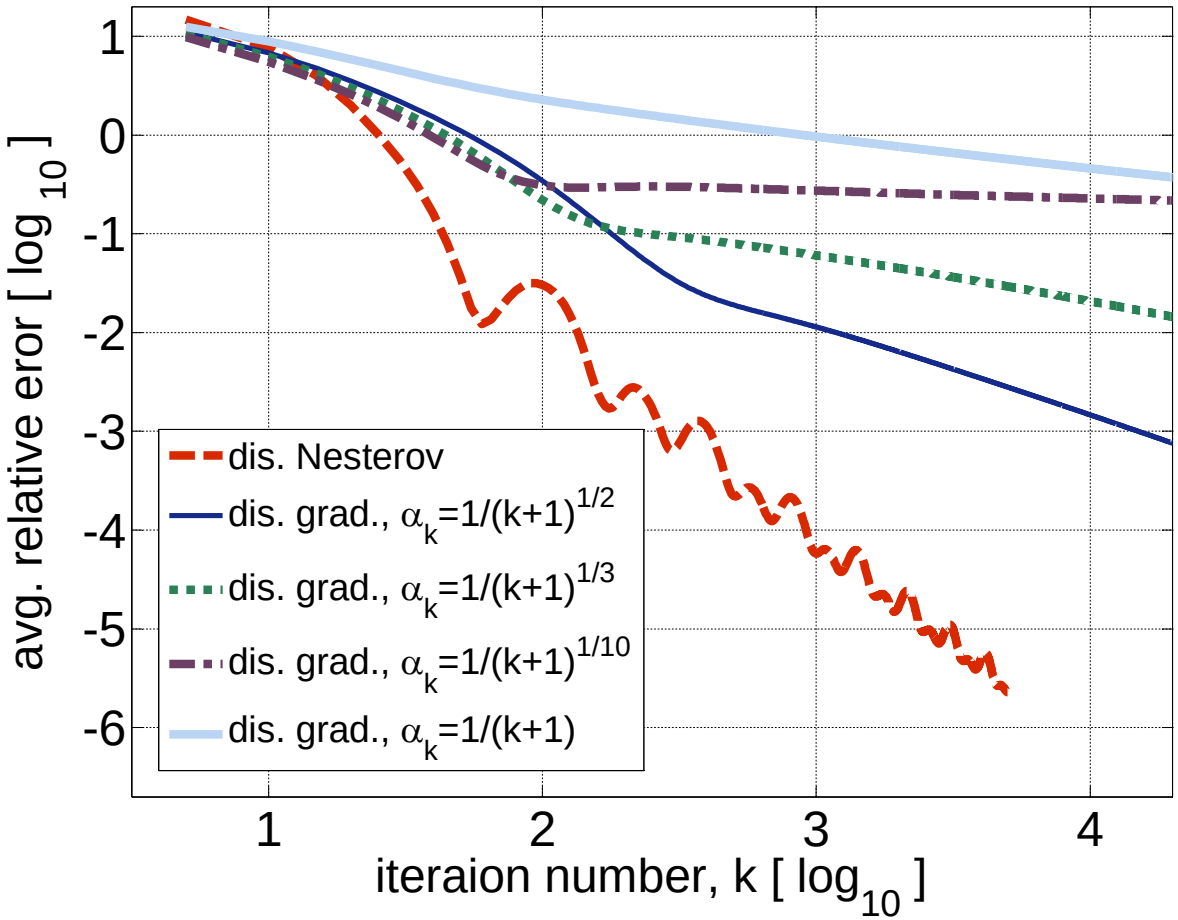
  with $C_i = \left\{x \,:\, \|x - r_i\| \leq \frac{1}{\sqrt{y_i}}\right\}$

- geometric graph, $n = 70$ nodes and 299 edges

---

[5] A. O. Hero and D. Blatt, "Sensor network source localization via projection onto convex sets (POCS)", *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2005*

$$e_f(t) := \frac{1}{n}\sum_{i=1}^{n} \frac{f(x_i(t)) - f^\star}{f^\star}$$



Red: D-NG[6], $\alpha(t) = 1/(t+1)$ Others: subgradient[7]

---

[6] D. Jakovetić *et al.*, "Fast cooperative distributed learning", 46*th Asilomar Conference on Signals, Systems and Computers (ASILOMAR)*, 2012

[7] A. Nedić and A. Ozdaglar, "Distributed subgradient methods for multi-agent optimization," *IEEE TAC*, 54(1), 2009

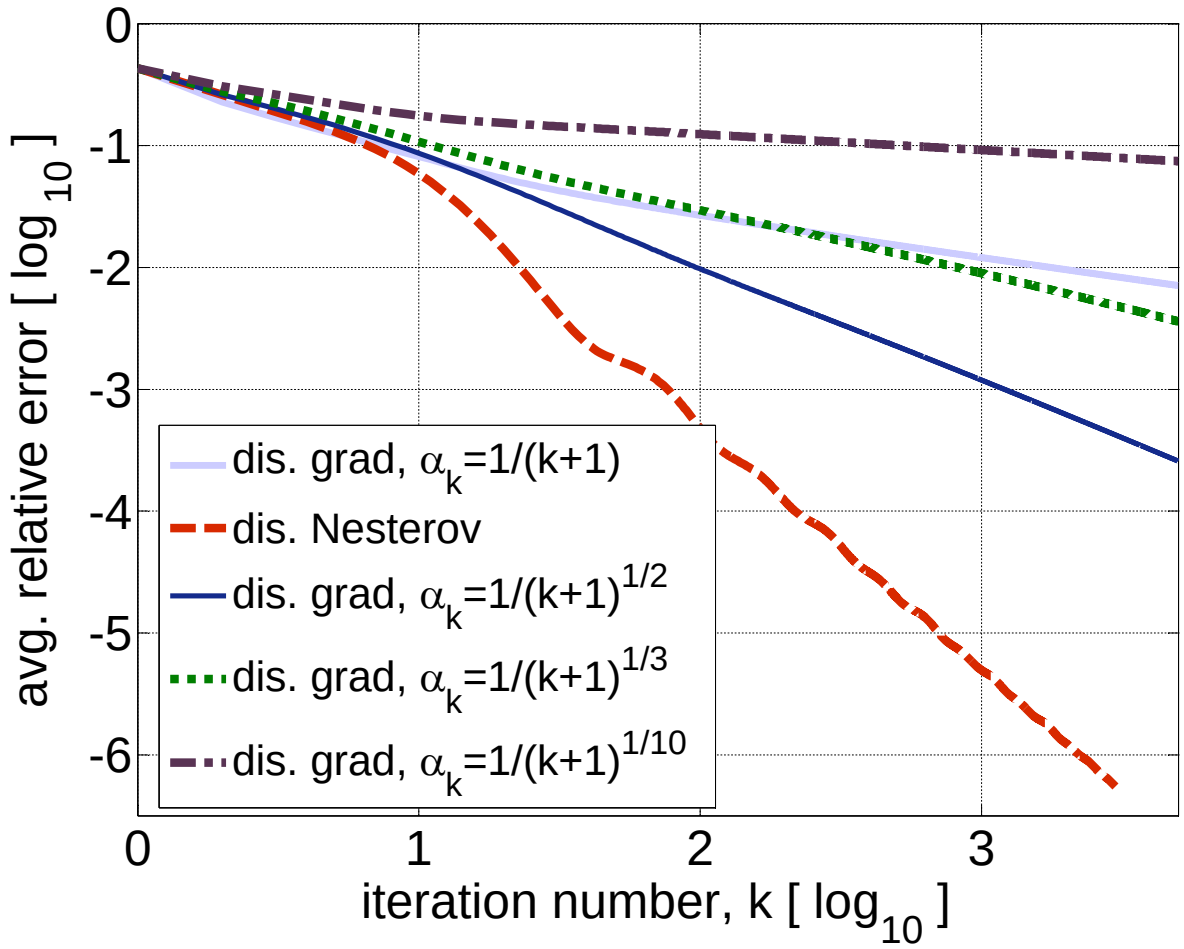
Numerical example

Distributed regularized logistic regression

$$\underset{s,r}{\text{minimize}} \quad \sum_{i=1}^{n} \underbrace{\phi\left(-b_i(s^\top a_i + r)\right) + \beta \|s\|^2}_{f_i(s,r)}$$

- (a_i, b_i): training data for agent i
- $\phi(t) = \log\left(1 + e^{-t}\right)$
- geometric graph, $n = 20$ nodes and 67 edges

$$e_f(t) := \frac{1}{n} \sum_{i=1}^{n} \frac{f(x_i(t)) - f^\star}{f^\star}$$



Red: D-NG, $\alpha(t) = 1/(t+1)$ Others: subgradient

Distributed Nesterov gradient method (D-NC) with consensus iterations

$$
\begin{aligned}
x(t) &= W^{a(t)}\left(y(t-1) - \alpha \nabla F(y(t-1))\right) \\
y(t) &= W^{b(t)}\left(x(t) + \frac{t-1}{t+2}\left(x(t) - x(t-1)\right)\right)
\end{aligned}
$$

- $a(t) = \lceil \frac{2 \log t}{-\log |\lambda|_2(W)} \rceil$ and $b(t) = \lceil \frac{\log 3}{-\log |\lambda|_2(W)} + \frac{2 \log t}{-\log |\lambda|_2(W)} \rceil$
- $|\lambda|_2(W)$ must be known by all agents

D. Jakovetić *et al.*, "Fast distributed gradient methods", arXiv preprint, 2011

Convergence analysis: if f_i's are differentiable, ∇f_i's are bounded and L-Lipschitz continuous, W is symmetric and stochastic, and $\alpha = 1/(2L)$

$$f(x_i(t)) - f^\star = \mathcal{O}\left(\frac{1}{t^2}\right)$$

- t iterations involve $\mathcal{O}(t \log t)$ communication rounds
- dependence on network spectral gap also known

Similar rate guarantees in:

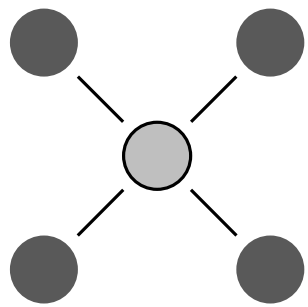
- A. Chen and A. Ozdaglar, "A fast distributed proximal gradient method," 50*th Allerton Conference on Communication, Control and Computing*, 2012

Distributed optimization via ADMM (D-ADMM)

ADMM = Alternate Direction Method of Multipliers

Recent review:

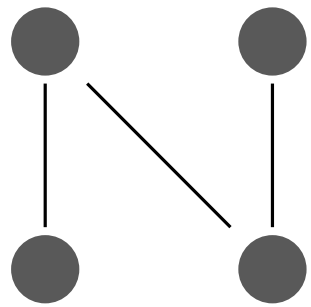
- S. Boyd *et al.*, "Distributed optimization and statistical learning via the alternating method of multipliers," *Foundations and Trends in Machine Learning*, 2011



Star network

ADMM, distributed optimization:

- I. Schizas *et al.*, "Consensus in ad hoc WSNs with noisy links - part I: distributed estimation of deterministic signals," *IEEE TSP*, 56(1), 2008
- many others

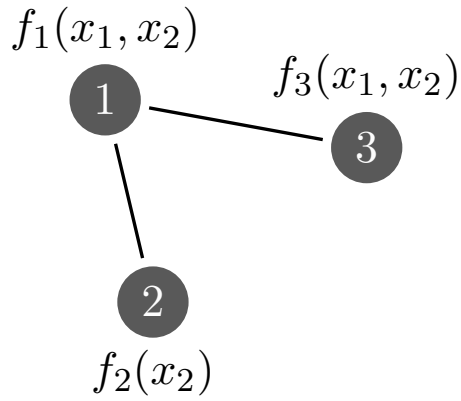


Generic network

This talk:

- J. Mota *et al.*, "D-ADMM: a communication efficient distributed algorithm for separable optimization," *IEEE TSP*, 61(10), 2013
- J. Mota *et al.*, "Distributed optimization with local domains: applications in MPC and network flows," arXiv preprint, 2013

Illustrative example:



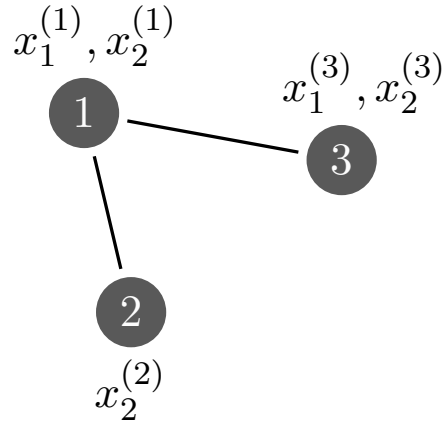
$$\underset{x_1,x_2,x_3}{\text{minimize}} \quad f_1(x_1, x_2) + f_2(x_2) + f_3(x_1, x_2)$$

- f_i's are proper, closed, convex functions with range $\mathbb{R} \cup \{+\infty\}$
- f_i's may depend on different subsets of variables

$$\underset{x_1,x_2,x_3}{\text{minimize}} \quad f_1(x_1,x_2) + f_2(x_2) + f_3(x_1,x_2)$$

Roadmap to obtain a distributed algorithm

Step 1: split each variable across associated nodes



Step 2: introduce consistency constraints

$$\begin{array}{ll} \text{minimize} & f_1\left(x_1^{(1)}, x_2^{(1)}\right) + f_2\left(x_2^{(2)}\right) + f_3\left(x_1^{(3)}, x_2^{(3)}\right) \\ \text{subject to} & x_1^{(1)} = x_1^{(3)} \quad x_2^{(1)} = x_2^{(3)} \\ & \qquad\qquad\quad x_2^{(1)} = x_2^{(2)} \end{array}$$

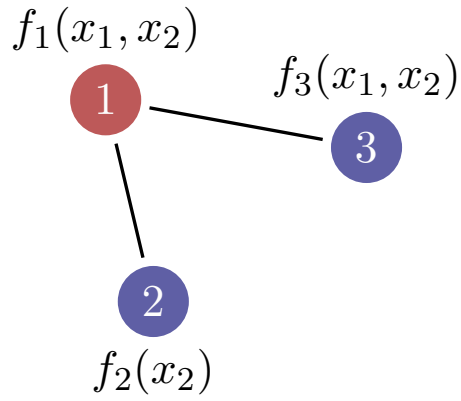
Step 3: pass to augmented Lagrangian dual

$$\text{maximize}_{\lambda_1^{(1,3)},\lambda_2^{(1,3)},\lambda_2^{(1,2)}} \quad \mathcal{L}_\rho\left(\lambda_1^{(1,3)},\lambda_2^{(1,3)},\lambda_2^{(1,2)}\right)$$

with

$$\begin{aligned}
\mathcal{L}_\rho\left(\lambda_1^{(1,3)},\lambda_2^{(1,3)},\lambda_2^{(1,2)}\right) \quad = \quad & f_1\left(x_1^{(1)},x_2^{(1)}\right)+f_2\left(x_2^{(2)}\right)+f_3\left(x_1^{(3)},x_2^{(3)}\right) \\
& +\langle\lambda_1^{(1,3)},x_1^{(1)}-x_1^{(3)}\rangle+\frac{\rho}{2}\left\|x_1^{(1)}-x_1^{(3)}\right\|^2 \\
& +\langle\lambda_2^{(1,3)},x_2^{(1)}-x_2^{(3)}\rangle+\frac{\rho}{2}\left\|x_2^{(1)}-x_2^{(3)}\right\|^2 \\
& +\langle\lambda_2^{(1,2)},x_2^{(1)}-x_2^{(2)}\rangle+\frac{\rho}{2}\left\|x_2^{(1)}-x_2^{(2)}\right\|^2
\end{aligned}$$

Step 4: color the network



$$
\begin{aligned}
\mathcal{L}_\rho\left(\lambda_1^{(1,3)}, \lambda_2^{(1,3)}, \lambda_2^{(1,2)}\right) \quad = \quad & f_1\left(x_1^{(1)}, x_2^{(1)}\right) + f_2\left(x_2^{(2)}\right) + f_3\left(x_1^{(3)}, x_2^{(3)}\right) \\
& + \langle \lambda_1^{(1,3)}, x_1^{(1)} - x_1^{(3)} \rangle + \frac{\rho}{2} \left\| x_1^{(1)} - x_1^{(3)} \right\|^2 \\
& + \langle \lambda_2^{(1,3)}, x_2^{(1)} - x_2^{(3)} \rangle + \frac{\rho}{2} \left\| x_2^{(1)} - x_2^{(3)} \right\|^2 \\
& + \langle \lambda_2^{(1,2)}, x_2^{(1)} - x_2^{(2)} \rangle + \frac{\rho}{2} \left\| x_2^{(1)} - x_2^{(2)} \right\|^2
\end{aligned}
$$

Step 5: apply extended ADMM

- Primal update at node 1

$$\begin{aligned}\left(x_1^{(1)}, x_2^{(1)}\right)(t+1) \quad = \quad & \operatorname{argmin}_{x_1,x_2} f_1(x_1, x_2) \\ & + \langle \lambda_1^{(1,3)}(t), x_1 \rangle + \frac{\rho}{2} \left\| x_1 - x_1^{(3)}(t) \right\|^2 \\ & + \langle \lambda_2^{(1,3)}(t), x_2 \rangle + \frac{\rho}{2} \left\| x_2 - x_2^{(3)}(t) \right\|^2 \\ & + \langle \lambda_2^{(1,2)}(t), x_2 \rangle + \frac{\rho}{2} \left\| x_2 - x_2^{(2)}(t) \right\|^2\end{aligned}$$

- Primal update at node 2

$$
\begin{aligned}
x_2^{(2)}(t+1) \quad = \quad & \operatorname{argmin}_{x_2} f_2(x_2) \\
& -\langle \lambda_2^{(1,2)}(t), x_2 \rangle + \frac{\rho}{2} \left\| x_2^{(1)}(t+1) - x_2 \right\|^2
\end{aligned}
$$

- Primal update at node 3

$$
\begin{aligned}
\left(x_1^{(3)}, x_2^{(3)} \right)(t+1) \quad = \quad & \operatorname{argmin}_{x_1, x_2} f_3(x_1, x_2) \\
& -\langle \lambda_1^{(1,3)}(t), x_1 \rangle + \frac{\rho}{2} \left\| x_1^{(1)}(t+1) - x_1 \right\|^2 \\
& -\langle \lambda_2^{(1,3)}(t), x_2 \rangle + \frac{\rho}{2} \left\| x_2^{(1)}(t+1) - x_2 \right\|^2
\end{aligned}
$$

Key point: nodes 2 and 3 work in parallel (same color)

Step 6: dual update at all relevant nodes

$$\begin{aligned}
\lambda_1^{(1,3)}(t+1) &= \lambda_1^{(1,3)}(t) + \rho\left(x_1^{(1)}(t+1) - x_1^{(3)}(t+1)\right) \\
\lambda_2^{(1,3)}(t+1) &= \lambda_2^{(1,3)}(t) + \rho\left(x_2^{(1)}(t+1) - x_2^{(3)}(t+1)\right) \\
\lambda_2^{(1,2)}(t+1) &= \lambda_2^{(1,2)}(t) + \rho\left(x_2^{(1)}(t+1) - x_2^{(2)}(t+1)\right)
\end{aligned}$$

Convergence analysis:

- for 2 colors (bipartite network): classical ADMM results apply
- for ≥ 3 colors[8]: convergence for strongly convex functions and suitable ρ

[8] D. Han and X. Yuan, "A note on the alternating direction method of multipliers," JOTA, 155(1), 2012

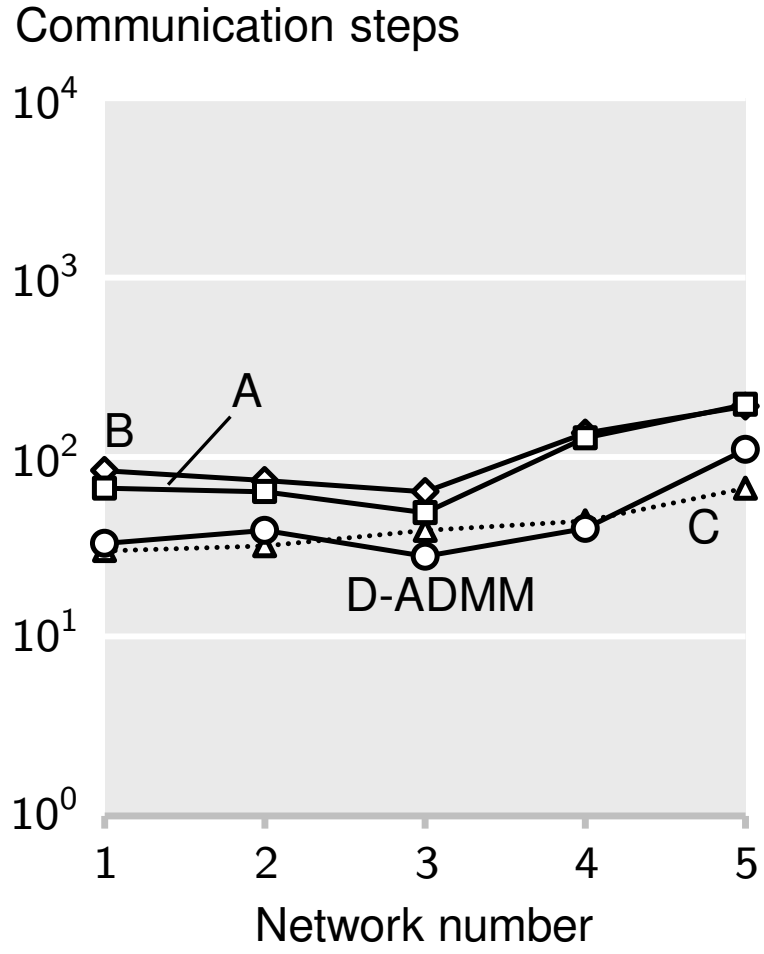
Numerical example

Consensus

$$\underset{x}{\text{minimize}} \quad \underbrace{\sum_{i=1}^{n}(x-\theta_i)^2}_{f_i(x)}$$

- $\theta_i =$ measurement of agent i ($\theta_i \overset{\text{i.i.d.}}{\sim} \mathcal{N}\left(10, 10^4\right)$)

Network	Model (parameters)	# Colors
1	Erdős-Rényi (0.12)	5
2	Watts-Strogatz $(4, 0.4)$	4
3	Barabasi (2)	3
4	Geometric (0.23)	10
5	Lattice (5×10)	2



D-ADMM[9] Others: A[10], B[11], C[12]

[9] J. Mota *et al.*, "D-ADMM: a communication efficient distributed algorithm for separable optimization," *IEEE TSP*, 61(10), 2013

[10] I. Schizas *et al.*, "Consensus in ad hoc WSNs with noisy links - part I: distributed estimation of deterministic signals," *IEEE TSP*, 56(1), 2008

[11] H. Zhu *et al.*, "Distributed in-network channel decoding," *IEEE TSP*, 57(10), 2009

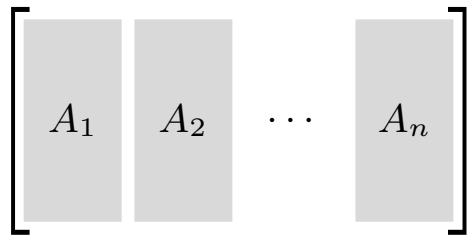
[12] B. Oreshkin *et al.*, "Optimization and analysis of distributed averaging with short node memory," *IEEE TSP*, 58(5), 2010

Numerical example

LASSO

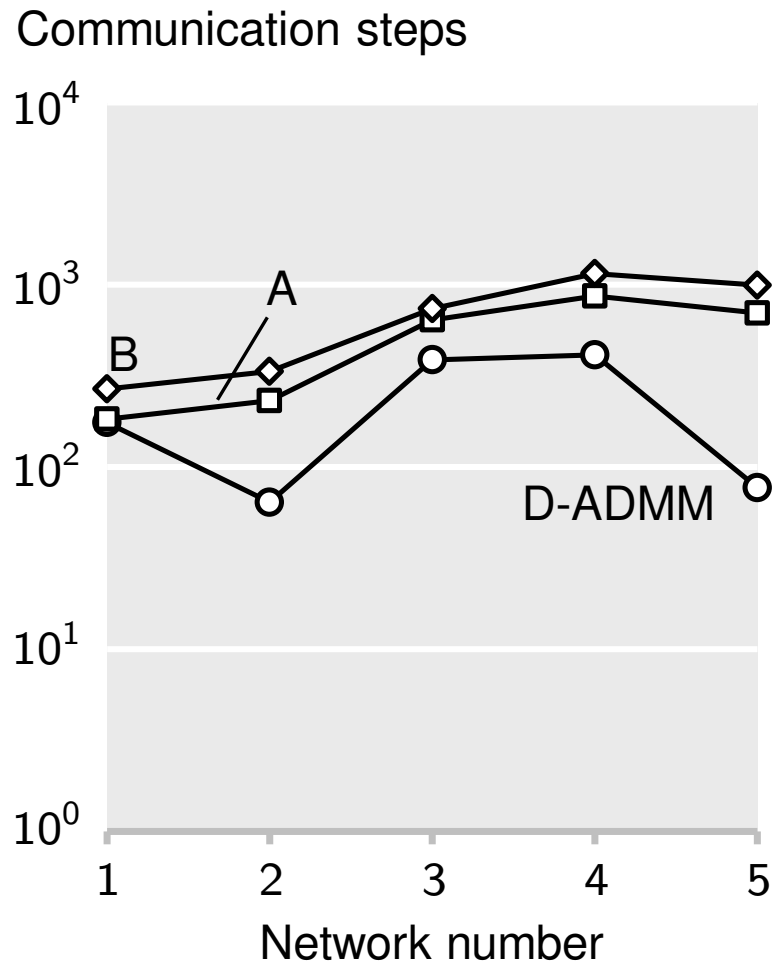
$$\begin{array}{ll} \underset{x}{\text{minimize}} & \|x\|_1 \\ \text{subject to} & \|Ax - b\| \leq \sigma \end{array}$$

Column partition: node i holds $A_i \in \mathbb{R}^{200\times 20}$



After regularization and dualization:

$$\underset{\lambda}{\text{minimize}} \quad \sum_{i=1}^{n} \underbrace{\frac{1}{n}(b^\top \lambda + \sigma \|\lambda\|) - \inf_x \left\{ \|x\|_1 + \lambda^\top A_i x + \frac{\delta}{2}\|x\|^2 \right\}}_{f_i(\lambda)}$$



D-ADMM[13] Others: A[14], B[15]

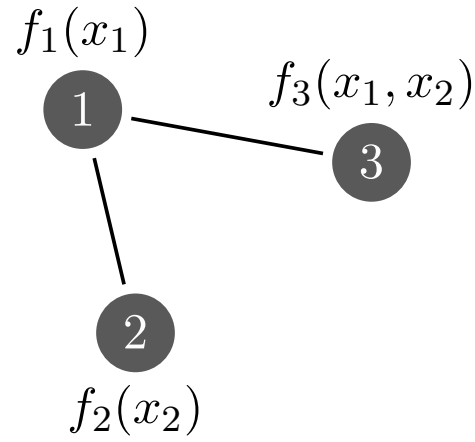
[13] J. Mota *et al.*, "D-ADMM: a communication efficient distributed algorithm for separable optimization," *IEEE TSP*, 61(10), 2013

[14] I. Schizas *et al.*, "Consensus in ad hoc WSNs with noisy links - part I: distributed estimation of deterministic signals," *IEEE TSP*, 56(1), 2008

[15] H. Zhu *et al.*, "Distributed in-network channel decoding," *IEEE TSP*, 57(10), 2009

What if variables are not "connected"?

Example: variable x_2 is not "connected"



Propagate variable across a Steiner tree[16]

[16] J. Mota *et al.*, "Distributed optimization with local domains: applications in MPC and network flows," arXiv preprint, 2013

Thank you!