Fast Algorithms for Distributed Large-Scale Optimization

Angelia.Nedich@asu.edu

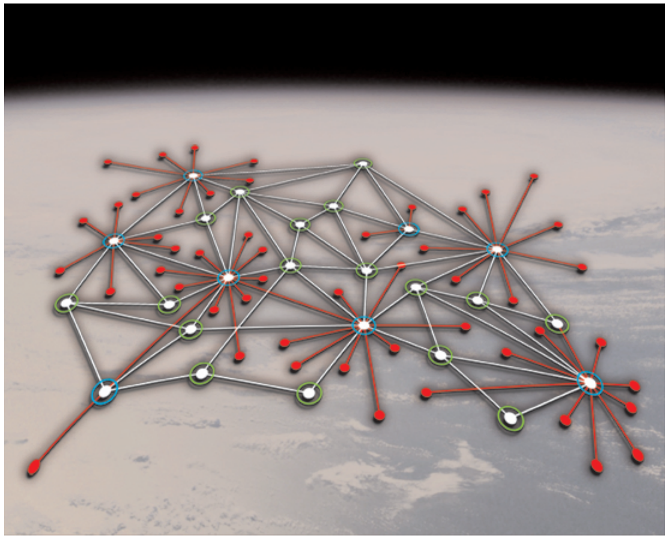
School of Electrical, Computer, and Energy Engineering

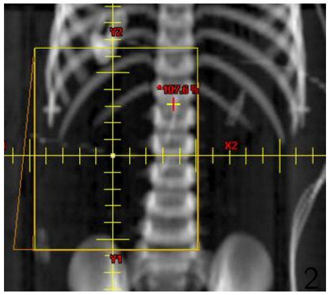
Arizona State University at Tempe

Collaborative work with

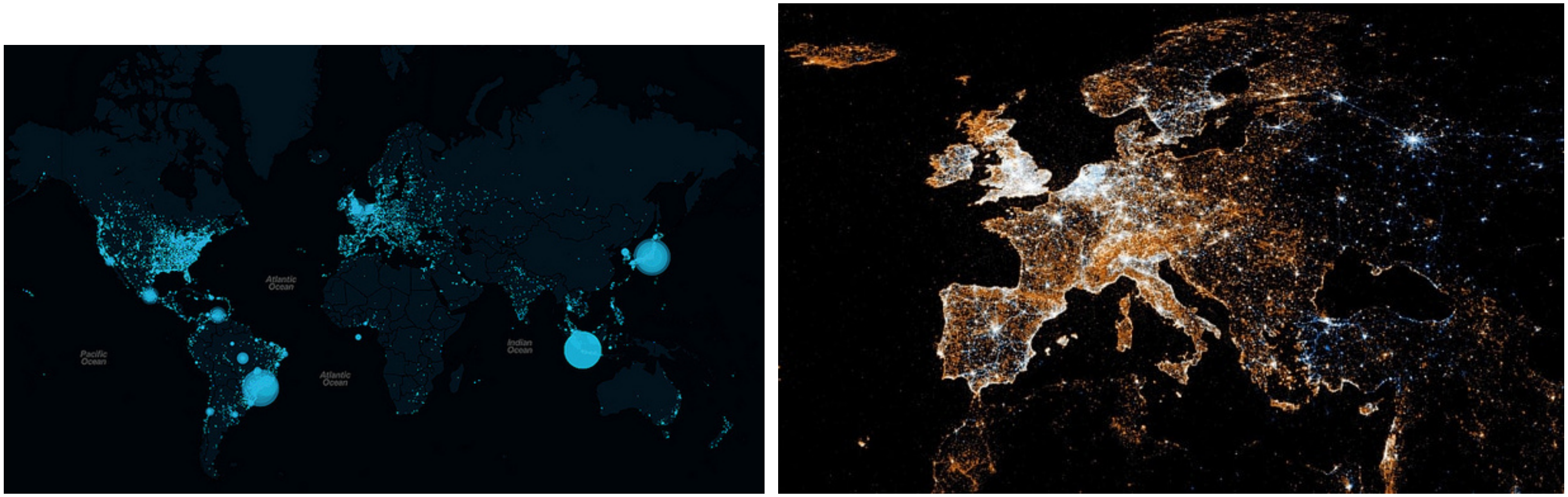
Wei (Wilbur) Shi and Alexander Olshevsky

Large-Scale Systems





Social Networks

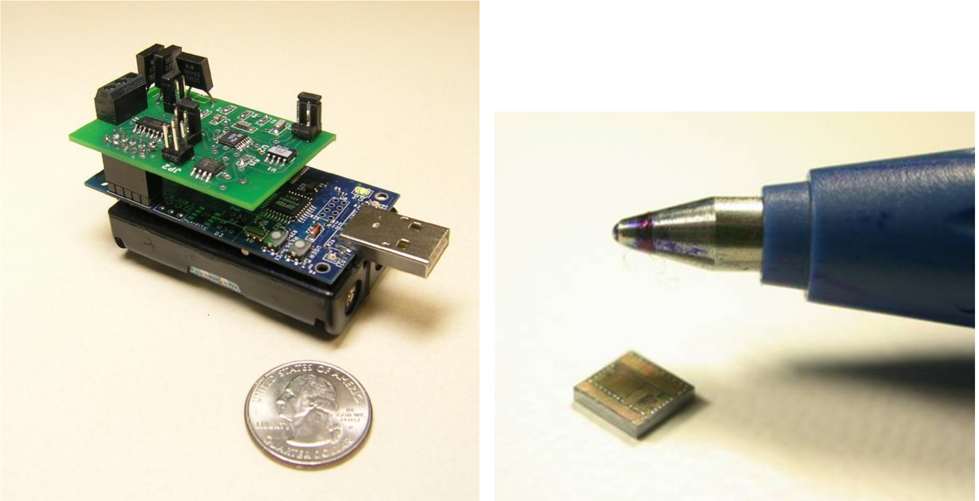


Left: Twitter Activity on New Year's Eve 2010[*], Image: *Twitter*
Right: Image of Twitter and Flicker activities over the past 3 years in Europe

Unprecedented ability to generate enormous amount of data in a second.
Need to process the data, learn from data, search through data, store the data etc., while in some domains the processing of the data should preserve "privacy".

[*] CDC 2010 was in Hilton, Atlanta, GA

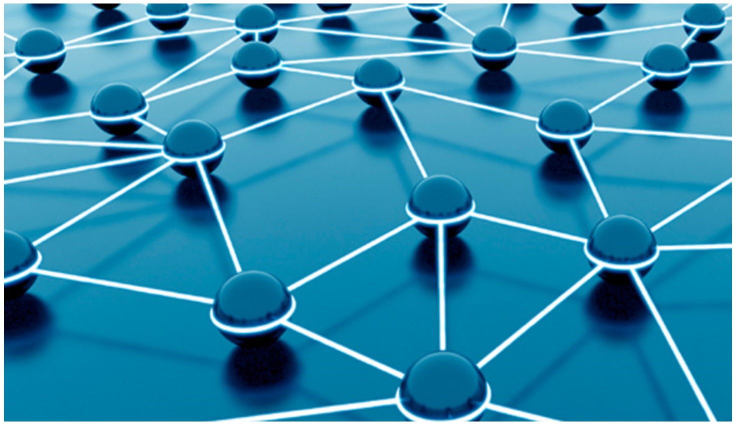
Sensor Networks



Left: A (large) mote compared to a nickel
Right: A mote with area about $5mm^2$ (from 2003), a dozen of these can fit in a penny. It contains all components of a mote: a CPU, memory, an A/D converter for reading sensor data, and a radio transmitter. It just needs a battery and an antenna to function.

As the sensors are getting smaller in size and less expensive, many new applications of sensor networks have emerged such as vehicle-sensor networks and body-sensor networks.

Sensor Networks $-- >$ Internet of Things



Wireless Sensor Networks (WSN), https://www.linkedin.com/pulse/internet-things-part-7-wireless-sensor-networks-mahendra-bhatia

“WSN technology applications for smart grid, smart water, intelligent transportation systems, and smart home generate huge amounts of data, ...
The term internet of things refers to uniquely identifiable objects and their virtual representations in an “internet-like” structure. These objects can be anything from large buildings, industrial plants, planes, cars, machines, any kind of goods.” [†]

[†] An article by Mahendra Bhatia at https://www.linkedin.com/pulse/internet-things-part-7-wireless-sensor-networks-mahendra-bhatia

Challenges: Requirements and System Characteristics

- Variety of operations/applications:
  - Detection, Identification, Estimation, Learning
  - Signal Processing, Communication
  - Data Processing: Storage and Retreival, Data Association, Data Mining, Clustering
  - Resource Allocation, Optimization and Control
- A wide range of performance requirements
  - Reliability, Robustness, Sustainability
  - Efficiency, Fairness
  - Security, Privacy
- Characteristics of the problems arising in the networked systems
  - Mobility, variability with time (not necessarily predictable)
  - Size (number of nodes/agents or number of the decision and/or constraints)

Agreement Model

Renewed interest in agreement problem by Vicsek 1995 Jadbabaie, Lin, Morse 2003

Literature:

Hegselmann and Krause 2002, Kempe, Dobra, and Gehrke 2003

Lin, Morse, Anderson 2003, 2004, Xiao and Boyd 2004, Moreau 2004, 2005

Olfati-Saber and Murray 2004, Lorenz 2005, Blondel, Hendrickx, Olshevsky, Tsitsiklis 2005

Cao, Spielman, Morse 2005, Boyd, Ghosh, Prabhakar, Shah 2005

Hatano, Das, and Mesbahi 2005, Ren and Beard 2005, Xiao, Boyd, and Lall 2005

Moallemi and Van Roy 2006, Carli, Fagnani, Speranzon, and Zampieri 2006

Nedić and Ozdaglar 2007, Marden, Arslan, and Shamma 2007

Kashyap, Başar, and Srikant 2007, Olfati-Saber, Fax, and Murray 2007

Patterson, Bamieh, and Abbadi 2007, Ren 2007, Xiao, Boyd, and Kim 2007

Huang and Manton 2007, 2008, Bliman and Ferrari-Trecate 2008

Bliman, Nedić, and Ozdaglar 2008, Cao, Morse, and Anderson 2008, Hendrickx 2008

Sundaram and Hadjicostis 2008, 2011, Olshevsky and Tsitsiklis 2008, 2009

Tahbaz-Salehi and Jadbabaie 2008, 2010, Patterson and Bamieh 2008, 2010

Aysal, Yildiz, Sarwate, and Scaglione 2009, Bullo, Corés, and Martínez 2009

Kar and Moura 2009, 2010, Nedić, Olshevsky, Ozdaglar, and Tsitsiklis 2009

Benezit, Blondel, Thiran, Tsitsiklis, Vetterli 2010, Carli, Fagnani, Frasca, Zampieri 2010

Dimakis, Kar, Moura, Rabbat, and Scaglione 2010, Olshevsky 2010, 2014

Zhu and S. Martínez 2010, Dominguez-Garcia and Hadjicostis 2011

Liu, Morse, Anderson, and Yu 2011, Cai and Ishii 2011

Lavaei and Murray 2012, Bolouki and Malhamé 2012, Sundaram, Revzen, Pappas 2012

Touri and Nedić 2009-2012, 2014, Touri 2012, Etesami and Başar 2013

Bajović, Xavier, Moura, and Sinopoli 2013, Hendrickx and Tsitsiklis 2013

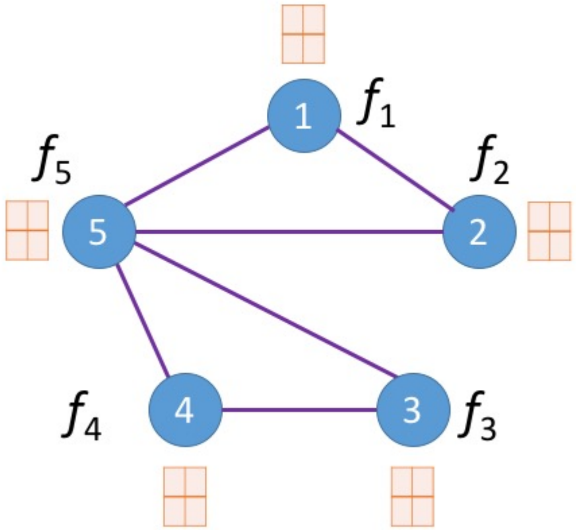
Mathkar and Borkar 2014, Başar, Etesami, and Olshevsky 2014

Borkar, Makhijani, and Sundaresan 2014, Touri and Langbort 2014, Bolouki 2014

Agreement and Optimization

- Suppose now each agent i has a local objective $f_i(x)$

- The agents are connected through an undirected graph $\mathbb{G}$ and can communicate locally
- Each agent can perform computations and has a buffer



- They need to cooperatively solve the following network problem

$$\text{minimize} \quad \sum_{i=1}^{m} f_i(x) \quad \text{subject to } x \in \mathbb{R}^n$$

  where each f_i is locally known to agent i only

- We assume that each f_i is convex and differentiable[‡]

[‡] For sake of discussion, convex and nondifferentiable will also work

- Assuming (for the moment) that the graph is static, connected and undirected
- Distributed and local consensus-based algorithm[§]

$$x_i(t+1) = \left(\sum_{j=1}^{m} a_{ij} x_j(t)\right) - \alpha_t \nabla f_i(x_i(t))$$

or

$$x_i(t+1) = \left(\sum_{j=1}^{m} a_{ij} x_j(t)\right) - \alpha_t \nabla f_i \left(\sum_{j=1}^{m} a_{ij} x_j(t)\right)$$

where $a_{ij} > 0$ if $j \in N_i \cup \{i\}$ and $a_{ij} = 0$ otherwise, and $\alpha_t > 0$ is a stepsize

Basic Convergence Result: assuming that the problem has a solution, the graph $\mathbb{G}$ is connected, the matrix A is doubly stochastic, the gradients are bounded and stepsize satisfies $\sum_{t=0}^{\infty} \alpha_t = +\infty$ and $\sum_{t=0}^{\infty} \alpha_t^2 < \infty$, one can show that

$$\lim_{t\to\infty} x_i(t) = x^* \qquad \text{for all } i$$

for an optimal solution x^*.

- In terms of time convergence the rate is of the order of $O(\frac{\ln t}{\sqrt{t}})$.
- If the function $\sum_{i=1}^{m} f_i(x)$ is strongly convex the rate is of the order of $O(\frac{\ln t}{t})$

[§] AN and A. Ozdaglar 2009

Algorithm Properties

- It is robust to network delays and other imperfections (e.g., missed messages)
- It can solve online problems, where the functions f_i may change with time
- It is efficient when dealing with (possibly stochastic) computational and/or communication errors
- Reliable and efficient in imperfect situations
- Extendable to variants for solving saddle-point problems and games

Work

AN, Olshevsky, Ozdaglar, and Tsitsiklis 2008 (with quantization effects)

Johansson, Rabi and M. Johansson 2009 (a randomized variant)

Ram, AN, Veeravalli 2009-2010, 2012 (various extensions)

Burger, Notarstefano, F. Bullo, and F. Allgöwer 2010 (distributed simplex)

AN, Ozdaglar, and Parrilo 2010 (with distributed constraints)

Cattivelli and Sayed 2010 (distributed estimation)

Wang and Elia 2011 (a control perspective)

Jakovetić, Xavier, and Moura 2011 (distributed Augmented Lagrangian)

Lobel and Ozdaglar 2011 (over random graphs)

Lobel, Ozdaglar, and Feijer 2011 (with state dependent weights)

Zanella, Varagnolo, Cenedese, Pillonetto, and Schenato 2011 (Newton-Raphson)

Chen and Sayed 2012, Lu and Tang 2012 (zero-gradient sum method)

Ram 2009, Srivastava 2011, Lee 2013, Wei (phD work on distributed optimization)

Zhu and Martínez 2012, 2013 (with constraints)

Ghadimi, Schame, Johansson 2013

Kvaternik 2014 (PhD work continuous model for distributed optimization)

Duchi, Agarwal, and Wainwright 2012 (distributed dual Nesterov method)
Li and Marden 2013 (designing games for distributed optimization)
Yan, Sundaram, Vishwanathan, and Qi 2013 (online)
Chang, AN, and Scaglione 2014 (distributed primal-dual perturbation method)
Gharesifard and Cortés 2012 (distributed continuous time model)
Xu 2016 (phD), Xu, Zhu, Soh, and Xie 2015 (augmented gradient methods)
Koshal, AN and Shanbhag 2016 (distributed algorithm for aggregative games)
AN, Lee, and Raginsky 2016 (online global objective minimization)
Notarnicolo and Notarstefano 2016, *Scaman, Bach Bubeck, Lee, Massoulié 2017*

Distributed ADMM Boyd, Parikh, Chu, Peleato, and Eckstein 2010
Ling and Ribeiro 2014, Wei and Ozdaglar 2012, 2013
Shi, Ling, Yuan, Wu, and Yin 2014
Aybat, Wang, Lin, and Ma 2015

Distributed Hypothesis Testing

Shahrampour and Jadbabaie 2013, Jadbabaie, Molavi, and Tahbaz-Salehi 2013, 2015
Shahrampour, Rakhlin, and Jadbabaie 2014, Lalitha, Javidi, and Sarwate 2014, 2015
AN, Olshevsky and Uribe 2015, 2016 Sahu and Kar 2016

Drawback: Balanced-Graph Requirement

$$x_i(t+1) = \left(\sum_{j=1}^{n} a_{ij} x_j(t)\right) - \alpha_t \nabla f_i(x_i(t))$$

- The matrix A **has to be doubly stochastic**, otherwise if only row-stochastic the algorithm would produce the iterates converging to a point that solves the problem

  $$\text{minimize} \quad \sum_{i=1}^{m} \pi_i f_i(x),$$

  where $\pi' A = \pi'$.

  The algorithm cannot be efficiently implemented in directed time-varying graphs[¶]

- An alternative to the weighted averaging is available through *push-sum algorithm* for consensus[*][‖]

[¶] Gharesifard and Cortés, "Distributed strategies for generating weight-balanced and doubly stochastic digraphs," European Journal of Control, 18 (6), 539–557, 2012

[‖] [*]D. Kempe, A. Dobra, and J. Gehrke *Gossip-based computation of aggregate information*, In Proceedings of the 44th Annual IEEE Symposium on Foundations of Computer Science, pages 482–491, 2003
F. Benezit, V. Blondel, P. Thiran, J. Tsitsiklis, and M. Vetterli *Weighted gossip: distributed averaging using non-doubly stochastic matrices*, In Proceedings of the 2010 IEEE International Symposium on Information Theory, 2010

Push-sum and Optimization methods

- Dominguez-Garcia and Hadjicostis. Distributed strategies for average consensus in directed graphs. In Proceedings of the IEEE Conference on Decision and Control, Dec 2011.
- Hadjicostis, Dominguez-Garcia, and Vaidya, "Resilient Average Consensus in the Presence of Heterogeneous Packet Dropping Links" CDC, 2012
- Tsianos and Rabbat. Distributed consensus and optimization under communication delays. In Proc. of Allerton Conference on Communication, Control, and Computing, 2011.
- Tsianos, Lawlor, and Rabbat. Consensus-based distributed optimization: Practical issues and applications in large-scale machine learning. In Proceedings of the 50th Allerton Conference on Communication, Control, and Computing, 2012.
- Tsianos, Lawlor, and Rabbat. Push-sum distributed dual averaging for convex optimization. In Proceedings of the IEEE Conference on Decision and Control, 2012.
- Tsianos. The role of the Network in Distributed Optimization Algorithms: Convergence Rates, Scalability, Communication / Computation Tradeoffs and Communication Delays. PhD thesis, McGill University, Dept. of Electrical and Computer Engineering, 2013.

Push-sum: Column Stochastic Matrix

- Given a directed and strongly connected graph $([m], E)$, let A be a matrix compatible with the graph
$$A_{ij} = 0 \qquad \text{when } (j,i) \notin E$$

- Assume that A has positive diagonal entries

- Also, let A be a column-stochastic matrix
$$A_{ij} \geq 0 \quad \text{for all } i,j \qquad \text{and} \qquad \mathbf{1}'A = \mathbf{1}'$$
where "prime" denotes the transpose and $\mathbf{1} = [1; \ldots; 1]$

- Then
$$\lim_{t\to\infty} A^t = \pi\mathbf{1}'$$
where π is a stochastic vector with $\pi_i > 0$ for all i

- Consider a process

$$x(t) = Ax(t-1) \qquad \text{for } t \geq 1$$

  with an arbitrary $x(0) \in \mathbb{R}^n$

- Then

$$\lim_{t\to\infty} x(t) = \lim_{t\to\infty} A^t x(0) = \pi \mathbf{1}' x(0) = \left(\mathbf{1}' x(0)\right) \pi$$

- Repeating this process with a different initial point, $y(0)$ we obtain

$$y(t) = Ay(t-1) \qquad \text{for } t \geq 1$$

$$\lim_{t\to\infty} y(t) = \left(\mathbf{1}' y(0)\right) \pi$$

- Look at the coordinate-wise ratio

$$z_i(t) = \frac{x_i(t)}{y_i(t)}, \qquad \lim_{t\to\infty} z_i(t) = \frac{(\mathbf{1}' x(0))\, \pi_i}{(\mathbf{1}' y(0))\, \pi_i} = \frac{\mathbf{1}' x(0)}{\mathbf{1}' y(0)}$$

- If we want

$$\lim_{t\to\infty} z_i(t) = \frac{1}{n} \mathbf{1}' x(0)$$

  it can be done by choosing the initial values $y_i(0) = 1$

Push-Sum Algorithm for Consensus

We given a directed graph $\mathbb{G}$

Every node i maintains scalar variable $x_i(t)$ and $y_i(t)$

These quantities will be updated by the nodes according to the rules,

$$
\begin{aligned}
x_i(t+1) &= \sum_{j \in N_i^{\rm in} \cup \{i\}} \frac{x_j(t)}{d_j+1}, \\
y_i(t+1) &= \sum_{j \in N_i^{\rm in} \cup \{i\}} \frac{y_j(t)}{d_j+1}, \\
z_i(t+1) &= \frac{x_i(t+1)}{y_i(t+1)}
\end{aligned}
$$

- Each node i "knows" its out degree d_i
- $N_i^{\rm in}$ is the set of "in"-neighbors of node i
- The method[†**] is initiated with $y_i(0) = 1$ for all i.

[**†] D. Kempe, A. Dobra, and J. Gehrke "Gossip-based computation of aggregate information" In Proceedings of the 44th Annual IEEE Symposium on Foundations of Computer Science, pages 482491, Oct. 2003
F. Benezit, V. Blondel, P. Thiran, J. Tsitsiklis, and M. Vetterli "Weighted gossip: distributed averaging using non-doubly stochastic matrices" In Proceedings of the 2010 IEEE International Symposium on Information Theory, Jun. 2010.

Perturbed Push-Sum: Scalar Case, Time-varying graphs

$$\begin{aligned}
w_i(t+1) &= \sum_{j\in N_i^{\mathrm{in}}(t)\cup\{i\}} \frac{x_j(t)}{d_j(t)+1}, \\
y_i(t+1) &= \sum_{j\in N_i^{\mathrm{in}}(t)\cup\{i\}} \frac{y_j(t)}{d_j(t)+1}, \\
z_i(t+1) &= \frac{w_i(t+1)}{y_i(t+1)} \\
x_i(t+1) &= w_i(t+1)+\epsilon_i(t+1) \qquad (1)
\end{aligned}$$

where $\epsilon_i(t+1)$ are perturbations experienced by node i

This allows for studying the (sub)gradient methods as a special perturbations

$$\epsilon_i(t+1) = \alpha_t \nabla f_i(z_i(t+1))$$

Convergence Result

Consider the sequences $\{z_i(t)\}$, $i = 1, \ldots, m$, generated by the push-sum method.

Lemma 1 (Key) *Assuming that the graph sequence $\{G(t)\}$ is B-uniformly strongly connected, for all $t \geq 1$ we have*

$$\left| z_i(t+1) - \frac{\sum_{i=1}^m x_i(t)}{m} \right| \leq \frac{8}{\delta} \left(\lambda^t \|x(0)\|_1 + \sum_{s=1}^{t} \lambda^{t-s} \|\epsilon(s)\|_1 \right),$$

where $\delta > 0$ and $\lambda \in (0,1)$ satisfy

$$\delta \geq \frac{1}{m^{mB}}, \qquad \lambda \leq \left(1 - \frac{1}{m^{mB}}\right)^{1/B}.$$

Define matrices $A(t)$ by $A_{ij}(t) = 1/(d_j(t)+1)$ for $j \in N_i^{\mathrm{in}}(t) \cup \{i\}$ and 0 otherwise
If each of the matrices $A(t)$ are doubly stochastic, then

$$\delta = 1, \qquad \lambda \leq \left(1 - \frac{1}{4m^3}\right)^{1/B}.$$

Optimization

The subgradient-push method can be used for minimizing $F(z) = \sum_{i=1}^{m} f_i(z)$ **over** $z \in \mathbb{R}^n$

Every node i maintains vectors $\mathbf{x}_i(t), \mathbf{w}_i(t)$ in $\mathbb{R}^n$, as well as an auxiliary scalar variable $y_i(t)$, initialized as $y_i(0) = 1$ for all i. These quantities will be updated by the nodes according to the rules,

$$\begin{aligned}
\mathbf{w}_i(t+1) &= \sum_{j \in N_i^{\text{in}}(t) \cup \{i\}} \frac{\mathbf{x}_j(t)}{d_j(t)+1}, \\
y_i(t+1) &= \sum_{j \in N_i^{\text{in}}(t) \cup \{i\}} \frac{y_j(t)}{d_j(t)+1}, \\
\mathbf{z}_i(t+1) &= \frac{\mathbf{w}_i(t+1)}{y_i(t+1)}, \\
\mathbf{x}_i(t+1) &= \mathbf{w}_i(t+1) - \alpha(t+1)\mathbf{g}_i(t+1), \qquad (2)
\end{aligned}$$

where $\mathbf{g}_i(t+1)$ is a subgradient of the function f_i at $\mathbf{z}_i(t+1)$.

The method is initiated with arbitrary $\mathbf{x}_i(0)$ and $y_i(0) = 1$ for all i.

The stepsize $\alpha(t+1) > 0$ satisfies the following decay conditions

$$\sum_{t=1}^{\infty} \alpha(t) = \infty, \qquad \sum_{t=1}^{\infty} \alpha^2(t) < \infty$$

Under this stepsize (and B-uniform strong connectivity), the algorithm produces the iterates that converge to a **consensual** minimizer of $F(z) = \sum_{i=1}^{m} f_i(z)$ over $z \in \mathbb{R}^n$.

- Simple broadcast-based implementation: each node i broadcasts the quantities $\mathbf{x}_i(t)/(d_i(t)+1), y_i(t)/(d_i(t)+1)$ to all of the nodes in its out-neighborhood[††], which simply sum all the messages they receive to obtain $\mathbf{w}_i(t+1)$ and $y_i(t+1)$.

- The update equations for $\mathbf{z}_i(t+1), \mathbf{x}_i(t+1)$ can then be executed without any further communications between nodes during step t.

- Convergence rate is of the order of $O(\ln t/\sqrt{t})$ for convex functions and $O(\ln t/t)$ for strongly convex functions[#‡‡]

- Tatarenko and Touri 2015 (Non-Convex Distributed Optimization)

[††] We note that we make use here of the assumption that node i knows its out-degree $d_i(t)$.

[‡‡] #AN and Olshevsky *Distributed Optimization over Time-varying Directed Graphs* IEEE Transactions on Automatic Control 60 (3) 601-615, 2015
AN and Olshevsky *Stochastic Gradient-Push for Strongly Convex Functions on Time-Varying Directed Graphs* arxiv 2015

Yet Another Issue

- The consensus-type algorithms discussed thus far will not produce convergent iterates when **a fixed stepsize** is used even when the functions f_i have Lipschitz gradients and $\sum_{i=1}^m f_i(x)$ is strongly convex

- Brought to attention in work of Shi, Ling, Wu, and Yin (EXTRA) 2014, 2015

$$x_i(t+1) = \left(\sum_{j=1}^m a_{ij} x_j(t)\right) - \alpha_t \nabla f_i(x_i(t))$$

$$x^* = x^* - \alpha \nabla f_i(x^*) \implies \nabla f_i(x^*) = 0$$

- No hope that the algorithms using the diminishing step can achieve a geometric rate!

- They are still good - work well in noisy environment - just the stepsize cannot be constant

Achieving Linear Rate: Methods that Track Gradients

- Centralized: $z(t+1) = z(t) + \sum_{i=1}^{m} \nabla f_i(z(t))$.

- **D**istributed **I**nexact **G**radient Track**ing** (DIGing)

  Basic Idea: use a "surrogate" for the sum of gradients

$$x_i(t+1) = \left(\sum_{j=1}^{m} a_{ij} x_j(t)\right) - \alpha_t \overset{g_i(t)}{\overbrace{\text{est. of } \textstyle\sum_{i=1}^{m} \nabla f_i(z(t))}}$$

$$g_i(t+1) = \left(\sum_{j=1}^{m} a_{ij} g_j(t)\right) + \nabla f_i(x_i(t+1)) - \nabla f_i(x_i(t))$$

  When the matrix A is doubly stochastic, it can be seen that

$$\sum_{i=1}^{m} g_i(t+1) = \sum_{i=1}^{m} \nabla f_i(x_i(t+1)).$$

- The convergence rate is geometric (variants for time-varying undirected and directed graphs)

- AN, Alex Olshevsky, Wei Shi *Achieving Geometric Convergence for Distributed Optimization over Time-Varying Graphs*, arxiv https://arxiv.org/abs/1607.03218

- AN, Alex Olshevsky, Wei Shi, Cesar Uribe *Geometrically Convergent Distributed Optimization with Uncoordinated Step-Sizes*, https://arxiv.org/pdf/1609.05877v1.pdf, 2016

Closely Related Literature and Simultaneous Work

- Tracking technique used in (not for gradients)
  M. Zhu and S. Martínez, *Discrete-Time Dynamic Average Consensus*, Automatica, 46, 2010,

- A method using gradient tracking proposed in
  J. Xu, S. Zhu, Y. Soh, and L. Xie, *Augmented Distributed Gradient Methods for Multi-Agent Optimization Under Uncoordinated Constant Stepsizes*, in Proceedings of the 54th IEEE Conference on Decision and Control (CDC), 2015, pp. 2055–2060.

- A part of Xu's thesis work
  J. Xu, *Augmented Distributed Optimization for Networked Systems*, PhD thesis, Nanyang Technological University, 2016.

- G. Qu and N. Li, *Harnessing Smoothness to Accelerate Distributed Optimization*, on arXiv at https://arxiv.org/abs/1605.07112, 2016.

Algorithm NEXT

- Work by Lorenzo and Scutari - considers general non-convex (objective) problems and a class of algorithms
  P. Di Lorenzo and G. Scutari *Distributed nonconvex optimization over networks*, in IEEE International Workshop on Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP), 2015, pp. 229–232.
  P. Di Lorenzo and G. Scutari, *NEXT: In-Network Nonconvex Optimization*, IEEE Transactions on Signal and Information Processing over Networks, 2016.
  P. Di Lorenzo and G. Scutari *Distributed nonconvex optimization over time-varying networks*, in IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2016, pp. 4124–4128.

Compact notation

- Each node i has variable $x_i \in \mathbb{R}^n$, placed on the ith row of a matrix $\mathbf{x}$.

$$\mathbf{x} \triangleq \begin{pmatrix} — & x_1^{\mathrm{T}} & — \\ — & x_2^{\mathrm{T}} & — \\ & \vdots & \\ — & x_m^{\mathrm{T}} & — \end{pmatrix} \in \mathbb{R}^{m\times n}.$$

- $\mathbf{x}$ is consensual if all rows are equal: $x_i^{\mathrm{T}} = x_j^{\mathrm{T}}, \forall i \neq j$.

$$\mathbf{f}(\mathbf{x}) \triangleq \sum_{i=1}^{m} f_i(x_i), \quad \nabla \mathbf{f}(\mathbf{x}) \triangleq \begin{pmatrix} — & (\nabla f_1(x_1))^{\mathrm{T}} & — \\ — & (\nabla f_2(x_2))^{\mathrm{T}} & — \\ & \vdots & \\ — & (\nabla f_m(x_m))^{\mathrm{T}} & — \end{pmatrix} \in \mathbb{R}^{m\times n}.$$

- original problem $\iff$ $\min \mathbf{f}(\mathbf{x}),\ \text{s.t. } x_i = x_j,\ \forall i \neq j$

DIGing Method for Undirected Graphs

DIGing: matrices $\mathbf{W}(k)$ are doubly stochastic

```
Choose step-size $\alpha > 0$ and pick any $\mathbf{x}(0) \in \mathbb{R}^{m\times n}$;
Initialize $\mathbf{y}(0) = \nabla\mathbf{f}(\mathbf{x}(0))$;
for $k = 0, 1, \cdots$ do
    $\mathbf{x}(k+1) = \mathbf{W}(k)\mathbf{x}(k) - \alpha\mathbf{y}(k)$;
    $\mathbf{y}(k+1) = \mathbf{W}(k)\mathbf{y}(k) + \nabla\mathbf{f}(\mathbf{x}(k+1)) - \nabla\mathbf{f}(\mathbf{x}(k))$;
end
```

Each agent i:

$$\begin{aligned} x_i(k+1) = {} & W_{ii}(k)x_i(k) + \textstyle\sum_{j\in\mathcal{N}_i^{\rm in}(k)} W_{ij}(k)x_j(k) - \alpha y_i(k); \\ y_i(k+1) = {} & W_{ii}(k)y_i(k) + \textstyle\sum_{j\in\mathcal{N}_i^{\rm in}(k)} W_{ij}(k)y_j(k) \\ & +\nabla f_i(x_i(k+1)) - \nabla f_i(x_i(k)). \end{aligned}$$

$\mathbf{W}(k)$ is compatible with the graph $\mathcal{G}(k)$: $W_{ij}(k) > 0$ when $\{i, j\} \in \mathcal{E}_k$ and $Wii(k) > 0$.

Assumptions for Linear Convergence Rate for DIGing

- The functions f_i are convex with Lipschitz gradients (with Lipschitz constant L_i)

- The sum $\frac{1}{m}\sum_{i=1}^{m} f_i$ is strongly convex

- The graphs $\mathcal{G}(k)$ are B-connected: for some integer $B \geq 1$, the graph $([m], \cup_{t=k}^{k+B-1}\mathcal{E}_t)$ is connected for all k.

- $\mathbf{W}(k)$ is doubly stochastic, compatible with the graph $\mathcal{G}(k)$, and there is a $\tau > 0$ such that for all k,

$$W_{ij}(k) \geq \tau \qquad \text{whenever} \quad W(k) > 0.$$

Under these assumptions we have the following result.

Theorem 1 (DIGing: Explicit geometric rate) *The sequence* $\{\mathbf{x}^k\}$ *generated by DIGing converges to the unique optimal solution* $\mathbf{x}^*$ *at a global R-linear rate* $O(\lambda^k)$*, where*

$$\lambda = \begin{cases} \sqrt[2B]{1 - \frac{\alpha\bar{\mu}}{1.5}}, & \textit{if } \alpha \in \left(0, \frac{1.5\left(\sqrt{J_1^2+(1-\delta^2)J_1}-\delta J_1\right)^2}{\bar{\mu}J_1(J_1+1)^2}\right], \\ \sqrt[B]{\sqrt{\frac{\alpha\bar{\mu}J_1}{1.5}} + \delta}, & \textit{if } \alpha \in \left(\frac{1.5\left(\sqrt{J_1^2+(1-\delta^2)J_1}-\delta J_1\right)^2}{\bar{\mu}J_1(J_1+1)^2}, \frac{1.5(1-\delta)^2}{\bar{\mu}J_1}\right], \end{cases}$$

$$\delta \triangleq \max_{k \geq B-1} \left\{\sigma_{\max}\left\{\mathbf{W}_B(k) - \frac{1}{m}\mathbf{1}\mathbf{1}^{\mathrm{T}}\right\}\right\} \ \textit{and } J_1 \triangleq 3\bar{\kappa}B^2\left(1 + 4\sqrt{m}\sqrt{\bar{\kappa}}\right),$$

for any step-size $\alpha \in \left(0, \frac{1.5(1-\delta)^2}{\bar{\mu}J_1}\right]$*, with* $\bar{\kappa} = \frac{L}{\bar{\mu}}$*,* $L = \max_i L_i$*.*

Push-DIGing for Directed graphs

Push-DIGing: matrices $\mathbf{C}(k)$ are column stochastic

```
Choose step-size $\alpha > 0$ and pick any $\mathbf{x}(0) = \mathbf{u}(0) \in \mathbb{R}^{m\times n}$;
Initialize $\mathbf{y}(0) = \nabla\mathbf{f}(\mathbf{x}(0))$, $\mathbf{v}(0) = \mathbf{1} \in \mathbb{R}^m$, and $\mathbf{V}(0) = \mathrm{diag}\,\{\mathbf{v}(0)\}$;
for $k = 0, 1, \cdots$ do
    $\mathbf{u}(k+1) = \mathbf{C}(k)(\mathbf{u}(k) - \alpha\mathbf{y}(k))$;
    $\mathbf{v}(k+1) = \mathbf{C}(k)\mathbf{v}(k)$;   $\mathbf{V}(k+1) = \mathrm{diag}\,\{\mathbf{v}(k+1)\}$;
    $\mathbf{x}(k+1) = (\mathbf{V}(k+1))^{-1}\mathbf{u}(k+1)$;
    $\mathbf{y}(k+1) = \mathbf{C}(k)\mathbf{y}(k) + \nabla\mathbf{f}(\mathbf{x}(k+1)) - \nabla\mathbf{f}(\mathbf{x}(k))$;
end
```

$\mathbf{C}(k)$ is compatible with the directed graph $\mathcal{G}(k)$: $C_{ij} = \frac{1}{1+d_j^{\mathrm{o}}(k)}$ when $(j,i) \in \mathcal{E}_k$ where $d_j^{\mathrm{o}}(k)$ is the out-degree of node j at time k.

Specialized result

Corollary 2 (DIGing: Polynomial networks scalability) *If the graph is undirected,* $\mathbf{W}(k)$ *is a lazy Metropolis matrix*

$$w_{ij}(k) = \begin{cases} 1/\left(1 + \max\{d_i(k), d_j(k)\}\right), & \textit{if } \{i,j\} \in \mathcal{E}_k, \\ 1 - \sum_{\ell \in \mathcal{N}(k)} W_{i\ell}(k), & \textit{if } j = i, \\ 0, & \textit{else}, \end{cases}$$

and the agents choose the step-size

$$\alpha = \frac{3(2/71)^2}{128 B^2 m^{4.5} L \sqrt{\bar{\kappa}}} - \frac{1.5}{\bar{\mu}} \left(\frac{(2/71)^2}{128 B^2 m^{4.5} \bar{\kappa}^{1.5}} \right)^2,$$

then to reach ε*-accuracy, the number of iterations needed by DIGing is*

$$O\left(B^3 m^{4.5} \bar{\kappa}^{1.5} \ln \frac{1}{\varepsilon} \right).$$

(polynomial scaling in directed graph is still open)

Difficulties in analysis

(i) Asymmetric operators $\mathbf{W}(k)$, $\mathbf{C}(k)$: no cocoercivity, even no monotonicity

(ii) Asymmetric operators $\mathbf{W}(k)$, $\mathbf{C}(k)$: hard to find a Lyapunov function

(iii) Time-varying graphs: may need time-varying metric

Main Proof Idea for DIGing

```
Choose step-size $\alpha > 0$ and pick any $\mathbf{x}(0) \in \mathbb{R}^{m\times n}$;
Initialize $\mathbf{y}(0) = \nabla \mathbf{f}(\mathbf{x}(0))$;
for $k = 0, 1, \cdots$ do
    $\mathbf{x}(k+1) = \mathbf{W}(k)\mathbf{x}(k) - \alpha \mathbf{y}(k)$;
    $\mathbf{y}(k+1) = \mathbf{W}(k)\mathbf{y}(k) + \nabla \mathbf{f}(\mathbf{x}(k+1)) - \nabla \mathbf{f}(\mathbf{x}(k))$;
end for
```

$\|\mathbf{q}(k)\|_{\mathrm{F}} = \|\mathbf{x}(k) - \mathbf{x}^*\|_{\mathrm{F}}$ (optimality residual)

$\|\mathbf{z}(k)\|_{\mathrm{F}} = \|\nabla \mathbf{f}(\mathbf{x}(k)) - \nabla \mathbf{f}(\mathbf{x}(k-1))\|_{\mathrm{F}}$ (gradient difference)

$\|\check{\mathbf{y}}(k)\|_{\mathrm{F}} = \|\mathbf{y}(k) - (1/m)\mathbf{1}\mathbf{1}^{\mathrm{T}}\mathbf{y}(k)\|_{\mathrm{F}}$ (consensus violation of $\mathbf{y}$)

$\|\check{\mathbf{x}}(k)\|_{\mathrm{F}} = \|\mathbf{x}(k) - (1/m)\mathbf{1}\mathbf{1}^{\mathrm{T}}\mathbf{x}(k)\|_{\mathrm{F}}$ (consensus violation of $\mathbf{x}$)

We show that the sequences are upper bounded geometrically $\mathbf{q} \to \mathbf{z} \to \check{\mathbf{y}} \to \check{\mathbf{x}} \to \mathbf{q}$

Small Gain Theorem

Theorem 3 *Suppose that* $\mathbf{s}^1, \ldots, \mathbf{s}^m$ *are sequences such that for all positive integers* K*, we have that* $\mathbf{s}^1 \to \mathbf{s}^2 \to \cdots \to \mathbf{s}^m \to \mathbf{s}^1$*:*

$$\|\mathbf{s}^{i+1}\|_{\mathrm{F}}^{\lambda,K} \leq \gamma_i \|\mathbf{s}^i\|_{\mathrm{F}}^{\lambda,K} + \omega_i \text{ for } i = 1, \cdots, m-1$$
$$\text{and } \|\mathbf{s}^1\|_{\mathrm{F}}^{\lambda,K} \leq \gamma_m \|\mathbf{s}^m\|_{\mathrm{F}}^{\lambda,K} + \omega_m$$

where the constants (gains) $\gamma_1, \ldots, \gamma_m$ *are nonnegative and satisfy* $\gamma_1\gamma_2\cdots\gamma_m < 1$*, (and the constants* ω_i*,* $\forall i$ *are bounded), then*

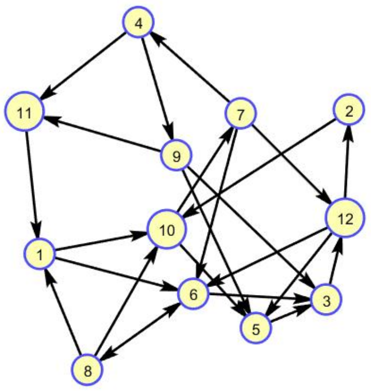
$$\|\mathbf{s}^i\|_{\mathrm{F}}^{\lambda,K} \leq \gamma_1\gamma_2\cdots\gamma_m \|\mathbf{s}^i\|_{\mathrm{F}}^{\lambda,K} + c_i, \ \forall i$$

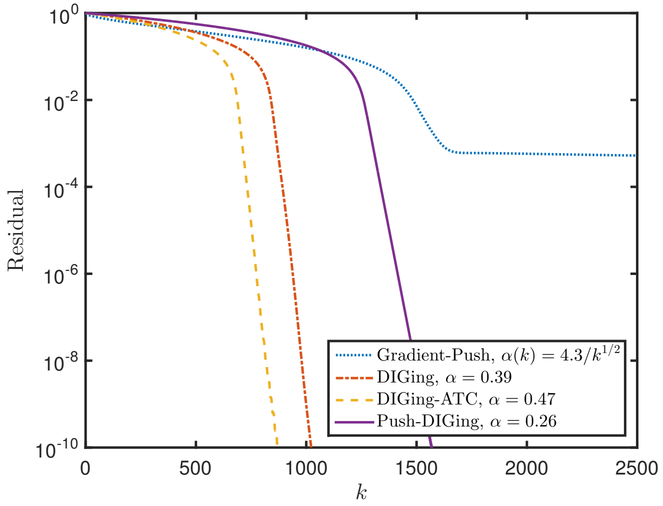
where for a sequence of matrices $\mathbf{s} = \{\mathbf{s}_0, \mathbf{s}_1, \ldots\}$

$$\|\mathbf{s}\|_{\mathrm{F}}^{\lambda,K} = \max_{0\leq k\leq K} \frac{1}{\lambda^k} \|s(k)\|_{\mathrm{F}}.$$

Numerical experiments: static directed graph

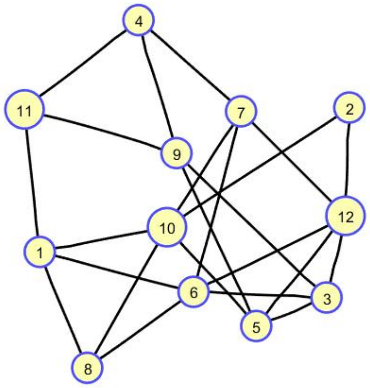
Each agent has a cost function given by a Huber loss. The DIGing is applied with a doubly stochastic matrix $\mathbf{W}$ (off line construction); DIGing and DIGing-ATC are fast

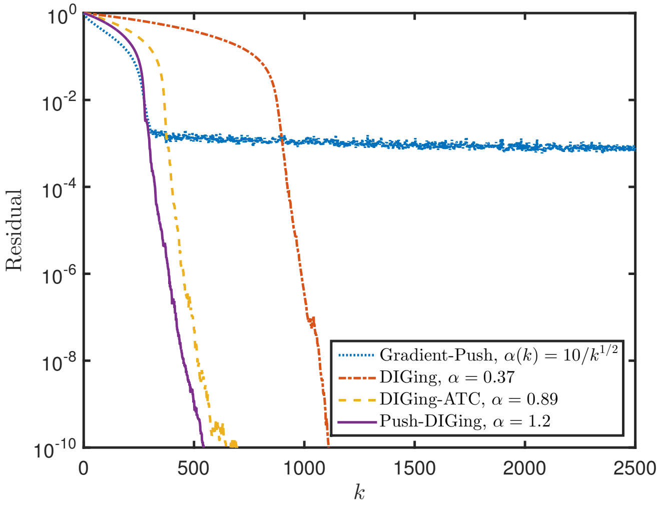




Numerical experiments: time-varying undirected graphs

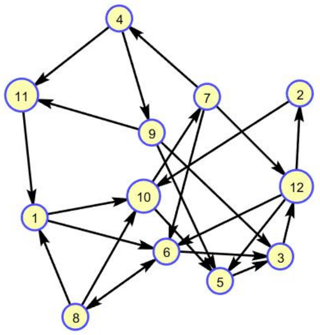
Time-varying graphs are generated randomly from a static graph (arc activation chance 40%) $\mathbf{W}(k)$ are Metropolis weights

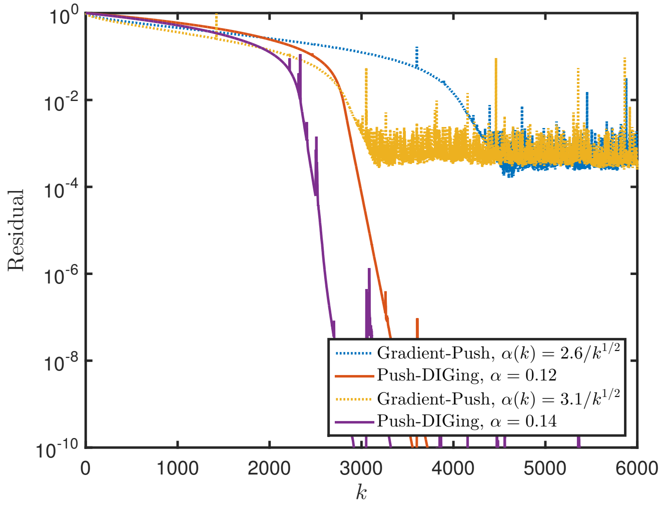




Numerical experiments: time-varying directed graphs

Time-varying graphs are generated randomly from a static graph (arc activation ratio 80% and then randomly choosing link direction)





Conclusions

- We have algorithms with linear convergence rate
- Theoretical bounds on the stepsize are "conservative" as the graphs are "general"
- Specializations to particular graph structure needed for practical purpose

AN, A. Olshevsky and W. Shi *Achieving Geometric Convergence For Distributed Optimization Over Time-Varying Graphs* arXiv:1607.03218