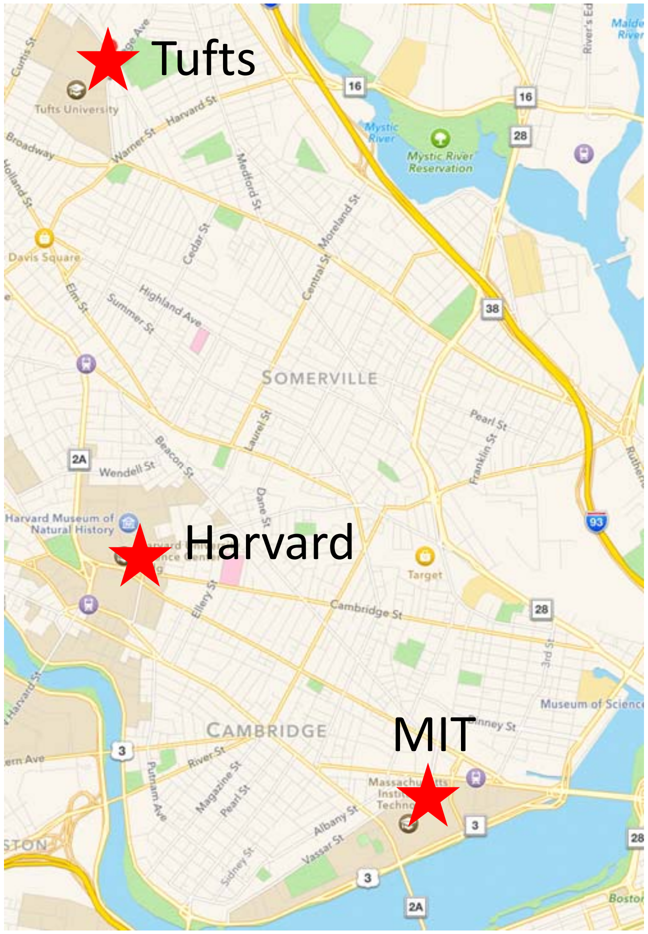
Distributed intelligence in multi-agent systems

Usman Khan
Department of Electrical and Computer Engineering
Tufts University

Workshop on Distributed Optimization, Information Processing, and Learning
Rutgers University
August 21, 2017

Who am I

- **Usman A. Khan**
  - **Associate Professor, Tufts**
- **Postdoc**
  - **U-Penn**
- **Education**
  - **PhD, Carnegie Mellon**
  - **MS, UW-Madison**
  - **BS, Pakistan**



Tufts University | School of Engineering
KTH Royal Institute of Technology

My Research Lab: Projects and demos

Research Team

PhD Students

Current

- Ran Kin, Sep. 2016 to date
- Fakhteh Saadatniaki, Sep. 2014 to date
- Sam Safavi, Jan. 2013 to date

Alumni

- Chengiua
  Distributed
- Mohamma
  Distributed

MS Studen

Current

- Noel Hwa
- Diane Lisk

Alumni

- Steven Sa
- Christoph
- Dong Park
- Alexander Henry, Aug. 2015, Adaptive methods for robotic path planning
- Michael Tran, Aug. 2015, Distributed target tracking in a sensor network
- Dibeyandu Das, Aug. 2015, Consensus with non-participating agents
- Anders Simpson-Wolf, Dec. 2014, Privacy and differentially private methods
- Luke Grymek, Aug. 2013, Coverage and surveillance with autonomous agents
- Gerald Solimini, May 2012,
- Syed S. Akbar, Dec. 2011,
- Qiong Wu (Applied Mathem … wind turbines

Undergraduates

Current

- Mathias Barth, Class of 2019, Jun. 2017 to date
- Isaac Collins, Class of 2020, Jun. 2017 to date
- Ashton Stevens, Class of 2019, Jun. 2017 to date
- Danie
- Mateo

Alum

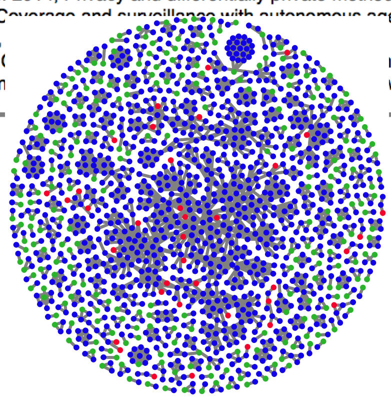
- Kathr
- Zacha
- Anuth
- Ryan
- Syed
- Craig
- Ryan
- Terrer
- Dong
- Cody
- Oghe
- Pratik … er 2014
- Corne

- Senior Design Project, 2014-2015
  - D. Park, C. Chen, and B. Zhang: GPS-based self-navigating UAV
- Senior Design Project, 2012-2013
  - … for
- Se … Drone
  - … e World
- ME
- Ke
- Ha
- Mi
- Je … 12
- Ty

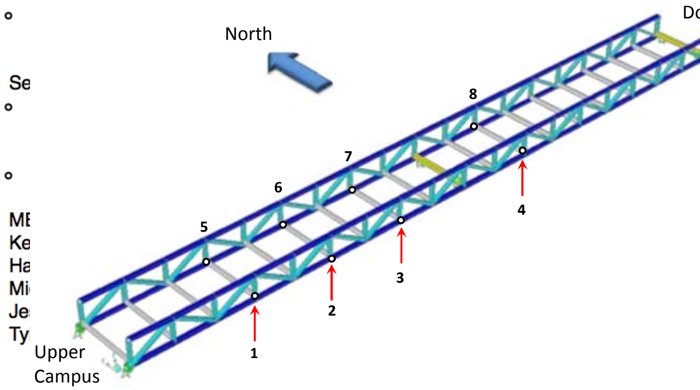
Inspecting leaks in NASA's lunar habitat

Aerial Formation Flying

Inference in Social Networks



SHM over a campus footbridge



Trailer

SPARTN—Signal Processing and RoboTic Networks Lab at Tufts

My Research Lab: Theory

Reza (2011-15):
Graph-theoretic estimation

Best paper
Journal cover

Xi (2012-16):
Optimization over directed graphs

4 TAC papers

Sam (2013-):
Fusion in non-deterministic graphs

2 Best papers
6 IEEE journal papers

Fakhteh (2014-):
Distributed estimation cont…d

Xin (2016-):
Optimization, Graph theory

My Research: In depth

- **Distributed Intelligence in multi-agent systems**
  - **Estimation, optimization, and control over graphs (networks)**

- **Mobile → Dynamic**
- **Heterogeneous → Directed**
- **Autonomous → Non-deterministic**

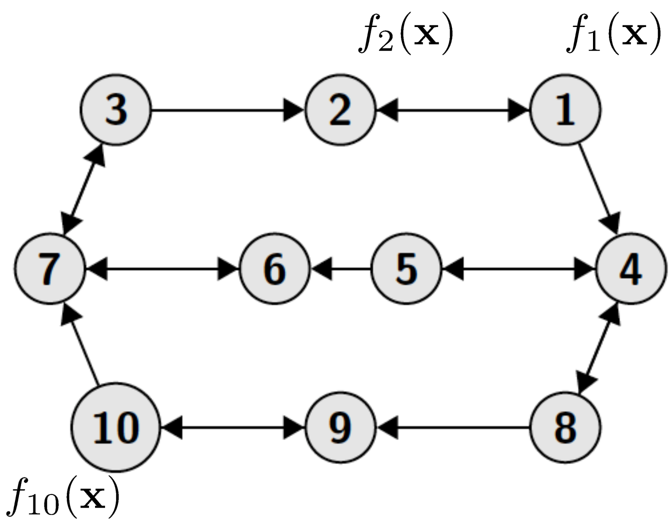
- **Applications:**
  - **Cyber-physical systems, IoTs, Big Data**
  - **Aerial SHM, Power grid, Personal exposome**
  - **Distributed Optimization: Path planning and Formation control**

Optimization over directed graphs

Problem

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

- **Agents interact over a graph**
  - **Directional informational flow**
- **No center with all information**



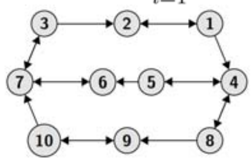
A nice solution

- **Gradient Descent**

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

  - **No one knows the function f**

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



- **Local Gradient Descent**

$$\mathbf{x}_{k+1}^i = \mathbf{x}_k^i - \alpha_k \nabla f_i(\mathbf{x}_k^i)$$

  - **Converges to only to a local optimal**

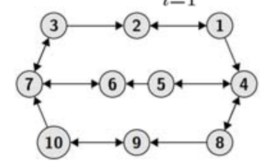
- **Distributed Gradient Descent [Nedich et al., 2009]: Fuse Information**

$$\mathbf{x}_{k+1}^i = \sum_{j \in \mathcal{N}_i} w_{ij} \mathbf{x}_k^j - \alpha_k \nabla f_i(\mathbf{x}_k^i)$$

Distributed Gradient Descent

- **Distributed Gradient Descent**

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



$$\begin{aligned} \text{Local:} \quad \mathbf{x}^i_{k+1} &= \sum_{j\in\mathcal{N}_i} w_{ij}\mathbf{x}^j_k - \alpha_k \nabla f_i(\mathbf{x}^i_k) \\ \text{Network:} \quad \mathbf{x}_{k+1} &= W\mathbf{x}_k - \alpha_k \nabla \mathbf{f}(\mathbf{x}_k) \end{aligned}$$

- $W=\{w_{ij}\}$ **is a doubly-stochastic matrix (underlying graph is balanced)**
- **Step-size goes to zero (but not too fast)**

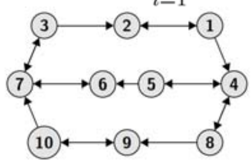
- **Agreement:** $W\mathbf{1} = \mathbf{1}$
- **Optimality:** $\mathbf{1}_n^\top \nabla \mathbf{f}_\infty = 0$

- **Lets do a simple analysis…**

Distributed Gradient Descent

- **Distributed Gradient Descent**

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



$$\begin{aligned} \text{Local:} \qquad \mathbf{x}_{k+1}^i &= \sum_{j\in\mathcal{N}_i} w_{ij}\mathbf{x}_k^j - \alpha_k \nabla f_i(\mathbf{x}_k^i) \\ \text{Network:} \qquad \mathbf{x}_{k+1} &= W\mathbf{x}_k - \alpha_k \nabla \mathbf{f}(\mathbf{x}_k) \end{aligned}$$

- **Assume the corresponding sequences converge to their limits**

$$\begin{aligned} \mathbf{x}_\infty &= W\mathbf{x}_\infty - \alpha_\infty \nabla \mathbf{f}(\mathbf{x}_\infty) \\ &= W\mathbf{x}_\infty - 0 \cdot \nabla \mathbf{f}(\mathbf{x}_\infty) \end{aligned} \longrightarrow (I_n - W)\mathbf{x}_\infty = \mathbf{0}$$

- **Let W be CS but not RS**
  $\mathbf{1}^\top W = \mathbf{1}^\top$ and $W\boldsymbol{\pi} = \boldsymbol{\pi} \neq \mathbf{1}$
- **Then** $\mathbf{x}_\infty = c\boldsymbol{\pi}$**, no agreement!**

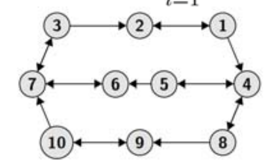
- **Let W be RS but not CS**
  $W\mathbf{1} = \mathbf{1}$ and $\boldsymbol{\pi}^\top W = \boldsymbol{\pi}^\top$
- **Then** $\mathbf{x}_\infty = c\mathbf{1}$**, i.e., agreement**
- **But suboptimal!**

$$\boldsymbol{\pi}^\top \nabla \mathbf{f}(\mathbf{x}_\infty) = \sum_{i=1}^{n} \pi_i \nabla f_i(c) = 0$$

Distributed Gradient Descent

- **Distributed Gradient Descent**

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



$$\begin{aligned} \text{Local:} \qquad \mathbf{x}_{k+1}^i &= \sum_{j\in\mathcal{N}_i} w_{ij}\mathbf{x}_k^j - \alpha_k \nabla f_i(\mathbf{x}_k^i) \\ \text{Network:} \qquad \mathbf{x}_{k+1} &= W\mathbf{x}_k - \alpha_k \nabla \mathbf{f}(\mathbf{x}_k) \end{aligned}$$

- **If W is RS but not CS (unbalanced directed graphs), agents agree on a suboptimal solution**

$$W\mathbf{1} = \mathbf{1} \text{ and } \boldsymbol{\pi}^\top W = \boldsymbol{\pi}^\top \qquad\qquad \boldsymbol{\pi}^\top \nabla \mathbf{f}(\mathbf{x}_\infty) = \sum_{i=1}^{n} \pi_i \nabla f_i(c) = 0$$

- **Consider a modification (Nedich 2013, similar in spirit but with different execution):**

$$\mathbf{x}_{k+1}^i = \sum_{j\in\mathcal{N}_i} w_{ij}\mathbf{x}_k^j - \alpha_k \frac{\nabla f_i(\mathbf{x}_k^i)}{[\underbrace{\mathbf{y}_k^i}_{\to\boldsymbol{\pi}}]_i}$$

- **Row-stochasticity guarantees agreement, scaling ensures optimality**
- **Estimate the left eigenvector?**

Tufts UNIVERSITY | School of Engineering

KTH ROYAL INSTITUTE OF TECHNOLOGY

Estimating the left eigenvector

- $A = \{a_{ij}\}$ **is row-stochastic with** $\boldsymbol{\pi}^\top A = \boldsymbol{\pi}^\top$
- **Consider the following iteration:**

$$\mathbf{y}_{k+1,i} = \sum_{j=1}^{n} a_{ij}\mathbf{y}_{k,j} \qquad \mathbf{y}_{0,i} = \mathbf{e}_i$$

$$Y_{k+1} = \begin{bmatrix} \mathbf{y}_{k+1,1}^\top \\ \vdots \\ \mathbf{y}_{k+1,n}^\top \end{bmatrix} = \begin{bmatrix} \sum_{j=1}^n a_{1j}\mathbf{y}_{k,j}^\top \\ \vdots \\ \sum_{j=1}^n a_{nj}\mathbf{y}_{k,j}^\top \end{bmatrix} = \begin{bmatrix} a_{11}\mathbf{y}_{k,1}^\top + \cdots + a_{1n}\mathbf{y}_{k,n}^\top \\ \vdots \\ a_{n1}\mathbf{y}_{k,1}^\top + \cdots + a_{nn}\mathbf{y}_{k,n}^\top \end{bmatrix} = AY_k$$

$$Y_\infty \triangleq \lim_{k\to\infty} Y_{k+1} = A^\infty Y_0 = A^\infty I_n = \boxed{A^\infty = \mathbf{1}_n\boldsymbol{\pi}^\top}$$

- **Every agent learns the entire left eigenvector asymptotically**
- **Similar method learns the right eigenvector for CS matrices**

Optimization over directed graphs: Recipe

- 1. Design row- or column-stochastic weights
- 2. Estimate the non-1 eigenvector for the eval of 1
- 3. Scale to remove the imbalance

- Side note: Push-sum algorithm (Gehrke et al., 2003; Vetterli et al., 2010)

Related work (a very small sample)

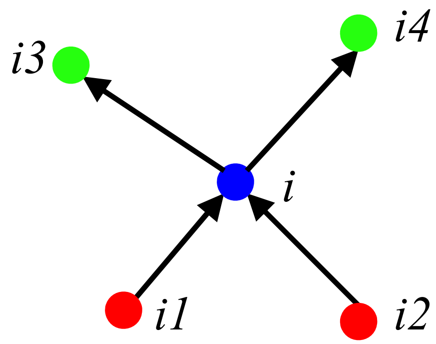
- **Algorithms over undirected graphs:**
  - **Distributed Gradient Descent (Nedich et al., 2009)**
    - **Non-smooth**
  - **EXTRA (Yin et al., Apr. 2014)**
    - **Fuses information over past two iterates**
    - **Use gradient information over past two iterates**
    - **Smooth, Strong-convexity, Linear convergence**
  - **NEXT (Scutari et al., Dec. 2015)**
    - **Functions are smooth non-convex + non-smooth convex**
  - **Harnessing smoothness … (Li et al., May 2016)**
    - **Some similarities to EXTRA**

Related work (a small sample)

- **Add push-sum to the previous obtain algorithms for directed graphs:**
  - **Gradient Push (Nedich et al., 2013)**
    - **Sub-linear convergence**
  - **DEXTRA (Khan et al., Oct. 2015)**
    - **Strong-convexity, Linear convergence**
    - **Difficult to compute step-size interval**
  - **SONATA (Scutari et al., Jul. 2016)**
    - **Functions are (smooth non-convex + non-smooth convex)**
    - **Sub-linear convergence**
  - **ADD-OPT (Khan et al., Jun. 2016) and PUSH-DIGing (Nedich et al., Jul. 2016)**
    - **Strong-convexity, Linear convergence**
    - **Step-size interval lower bound is 0**

- **All these algorithms employ column-stochastic matrices**

Column- vs. Row-stochastic Weights



$$
\begin{array}{cc}
A = \begin{bmatrix} & & & a_{i_3 i} & \\ & & & 0 & \\ & & & 0 & \\ 0 & a_{ii_1} & 0 & a_{ii} & a_{ii_2} \\ & & & a_{i_4 i} & \end{bmatrix} & \begin{array}{l} \\ \\ \\ \leftarrow \text{incoming weights at } i \\ \\ \end{array} \\
\begin{array}{c} \uparrow \\ \text{outgoing} \end{array} &
\end{array}
$$

- **Incoming weights are simpler to design**
- **For column sum to be 1, agent i cannot design the incoming weights as it does not know the neighbors of i1 and i2**
  - **Column-stochastic weights thus are designed at outgoing edges**
  - **Requires the knowledge of out-neighbors or out-degree**

Optimization with Row-stochastic weights

- $A = \{a_{ij}\}$ **is row-stochastic**

$$\text{Left Eigenvector:} \quad \mathbf{y}_{k+1}^{i} = \sum_{j \in \mathcal{N}_i^{\text{in}}} a_{ij} \mathbf{y}_k^{i}, \qquad (\text{vector in } \mathbb{R}^n)$$

$$\text{Update:} \quad \mathbf{x}_{k+1}^{i} = \sum_{j \in \mathcal{N}_i^{\text{in}}} a_{ij} \mathbf{x}_k^{j} - \alpha \mathbf{z}_k^{i}$$

$$\mathbf{z}_{k+1}^{i} = \sum_{j \in \mathcal{N}_i^{\text{in}}} a_{ij} \mathbf{z}_k^{j} + \frac{\nabla f_i(\mathbf{x}_{k+1}^{i})}{y_{k+1}^{ii}} - \frac{\nabla f_i(\mathbf{x}_k^{i})}{y_k^{ii}}$$

- **Row-stochastic weight design is simple**
- **However, in contrast to CS methods:**
  - **Agents run an nth order consensus for the left eigenvector**
  - **Agents need unique identifiers**

Optimization with Row-stochastic weights

- $A = \{a_{ij}\}$ **is row-stochastic**
- **Vector form of the algorithm:** arbitrary $\mathbf{x}_0$, $\widetilde{Y}_0 = Y_0 = I_n$, and $\mathbf{z}_0 = \nabla f_0$

$$\begin{aligned} Y_{k+1} &= AY_k, \qquad Y_\infty = \mathbf{1}_n \boldsymbol{\pi}^\top \\ \mathbf{x}_{k+1} &= 2A\mathbf{x}_k - A^2\mathbf{x}_{k-1} - \alpha\left(\widetilde{Y}_k^{-1}\nabla\mathbf{f}_k - \widetilde{Y}_{k-1}\nabla\mathbf{f}_{k-1}\right) \end{aligned}$$

- **In contrast, with a column-stochastic B, ADDOPT/PUSH-DIGing is:**

$$\mathbf{x}_{k+1} = 2B\mathbf{x}_k - B^2\mathbf{x}_{k-1} - \alpha\left(\nabla\mathbf{f}(Y_k^{-1}\mathbf{x}_k) - \nabla\mathbf{f}_{k-1}(Y_{k-1}^{-1}\mathbf{x}_{k-1})\right)$$

  - **Iterate does not result in agreement**
  - **The function argument is scaled by the right eigenvector**
  - **Ensures optimality**

Optimization with Row-stochastic weights

- **Algorithm:** arbitrary $\mathbf{x}_0$, $\widetilde{Y}_0 = Y_0 = I_n$, and $\mathbf{z}_0 = \nabla f_0$

$$\begin{aligned} Y_{k+1} &= AY_k, \qquad Y_\infty = \mathbf{1}_n \boldsymbol{\pi}^\top \\ \mathbf{x}_{k+1} &= 2A\mathbf{x}_k - A^2\mathbf{x}_{k-1} - \alpha\left(\widetilde{Y}_k^{-1}\nabla \mathbf{f}_k - \widetilde{Y}_{k-1}^{-1}\nabla \mathbf{f}_{k-1}\right) \end{aligned}$$

- **A simple intuitive argument:**
- **Assume each sequence converges to its limit, then**

$$\mathbf{x}_\infty = 2A\mathbf{x}_\infty - A^2\mathbf{x}_\infty - \alpha\left(\widetilde{Y}_\infty^{-1}\nabla \mathbf{f}_\infty - \widetilde{Y}_\infty^{-1}\nabla \mathbf{f}_\infty\right)$$

$$(I_n - A)^2\mathbf{x}_\infty = \mathbf{0}$$

$$\mathbf{x}_\infty = c\mathbf{1}_n$$

- **Every agent agrees on** c

Optimization with Row-stochastic weights

- **Algorithm:** arbitrary $\mathbf{x}_0$, $\widetilde{Y}_0 = Y_0 = I_n$, and $\mathbf{z}_0 = \nabla f_0$

$$
\begin{aligned}
Y_{k+1} &= AY_k, \qquad Y_\infty = \mathbf{1}_n \boldsymbol{\pi}^\top \\
\mathbf{x}_{k+1} &= 2A\mathbf{x}_k - A^2\mathbf{x}_{k-1} - \alpha\left(\widetilde{Y}_k^{-1}\nabla \mathbf{f}_k - \widetilde{Y}_{k-1}^{-1}\nabla \mathbf{f}_{k-1}\right)
\end{aligned}
$$

- **Show that c is the optimal solution**
- **Sum the update over k:**

$$
\begin{aligned}
\alpha \widetilde{Y}_M^{-1}\nabla \mathbf{f}_M &= \sum_{k=0}^{M-1}(A-A^2)\mathbf{x}_r + A\mathbf{x}_M + \sum_{k=0}^{M}(A-I_n)\mathbf{x}_r - \mathbf{x}_{M+1}. \\
\alpha \boldsymbol{\pi}^\top \widetilde{Y}_M^{-1}\nabla \mathbf{f}_M &= \sum_{k=0}^{M-1}\cancel{\boldsymbol{\pi}^\top(A-A^2)}\mathbf{x}_r + \boldsymbol{\pi}^\top A\mathbf{x}_M + \sum_{k=0}^{M}\cancel{\boldsymbol{\pi}^\top(A-I_n)}\mathbf{x}_r - \boldsymbol{\pi}^\top\mathbf{x}_{M+1} \\
&= \boldsymbol{\pi}^\top A\mathbf{x}_M - \boldsymbol{\pi}^\top\mathbf{x}_{M+1}.
\end{aligned}
$$

Optimization with Row-stochastic weights

- **Algorithm:** arbitrary $\mathbf{x}_0$, $\widetilde{Y}_0 = Y_0 = I_n$, and $\mathbf{z}_0 = \nabla f_0$

$$
\begin{aligned}
Y_{k+1} &= AY_k, \qquad Y_\infty = \mathbf{1}_n \boldsymbol{\pi}^\top \\
\mathbf{x}_{k+1} &= 2A\mathbf{x}_k - A^2\mathbf{x}_{k-1} - \alpha\left(\widetilde{Y}_k^{-1}\nabla\mathbf{f}_k - \widetilde{Y}_{k-1}^{-1}\nabla\mathbf{f}_{k-1}\right)
\end{aligned}
$$

- **Asymptotically**

$$
\alpha\boldsymbol{\pi}^\top\widetilde{Y}_\infty^{-1}\nabla\mathbf{f}_\infty = \boldsymbol{\pi}^\top A\mathbf{x}_\infty - \boldsymbol{\pi}^\top\mathbf{x}_\infty
$$

$$
\alpha\mathbf{1}_n^\top\nabla\mathbf{f}_\infty = 0
$$

$$
\nabla f_1(x_\infty^1) + \nabla f_2(x_\infty^2) + \cdots \nabla f_n(x_\infty^n) = \nabla f_1(c) + \nabla f_2(c) + \cdots \nabla f_n(c) = 0
$$

Optimization with Row-stochastic weights

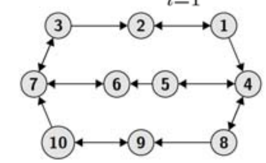
- **Algorithm:** arbitrary $\mathbf{x}_0$, $\widetilde{Y}_0 = Y_0 = I_n$, and $\mathbf{z}_0 = \nabla f_0$

$$
\begin{aligned}
Y_{k+1} &= AY_k, \qquad Y_\infty = \mathbf{1}_n \boldsymbol{\pi}^\top \\
\mathbf{x}_{k+1} &= 2A\mathbf{x}_k - A^2\mathbf{x}_{k-1} - \alpha\left(\widetilde{Y}_k^{-1}\nabla \mathbf{f}_k - \widetilde{Y}_{k-1}^{-1}\nabla \mathbf{f}_{k-1}\right)
\end{aligned}
$$

- **We assumed that the sequences reach their limit**
- **However, under what conditions and at what rate?**

Convergence conditions

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



- **Assume strong-connectivity, Lipschitz-continuous gradients, strongly-convex functions**

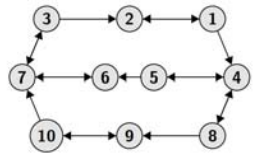
- **Consider** $\mathbf{t}_k = \begin{bmatrix} \|\mathbf{x}_k - \widehat{\mathbf{x}}_k\| \\ \|\widehat{\mathbf{x}}_k - \mathbf{x}^*\|_2 \\ \|\mathbf{z}_k - \widehat{\mathbf{z}}_k\| \end{bmatrix}$

- **If some norm of $\mathbf{t}_k$ goes to 0, then each element goes to 0 and the sequences converge to their limits**

Convergence conditions

$$\mathbf{x}_{k+1} \quad = \quad 2A\mathbf{x}_k - A^2\mathbf{x}_{k-1} - \alpha\left(\widetilde{Y}_k^{-1}\nabla\mathbf{f}_k - \widetilde{Y}_{k-1}^{-1}\nabla\mathbf{f}_{k-1}\right)$$

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



- **Assume strong-connectivity, Lipschitz-continuous gradients, strongly-convex functions**

- **Theorem 1.** *Let Assumptions A1 and A2 hold. We have*

$$\mathbf{t}_{k+1} \leq G\mathbf{t}_k + H_k\mathbf{s}_k, \qquad \forall k.$$

$$G_\alpha = \begin{bmatrix} \sigma & 0 & \alpha \\ \alpha cnl & 1-\alpha ns & 0 \\ c(\tau+\alpha nl) & \alpha d_1 ln & \sigma+\alpha c \end{bmatrix}$$

- **Lemma: H_k goes to 0 linearly**
- **Lemma: Spectral radius of G is less than 1**

Convergence conditions

- **Lemma:** For all values of $\alpha \in (0, \alpha_1)$, we have $\rho(G_\alpha) < 1$, where

$$\alpha_1 = \frac{\sqrt{\Delta^2 + 4cn^3 l(l+s)s(1-\sigma)^2} - \Delta}{2cn^2 l(l+s)} \text{ and } \Delta = cns(\tau + 1 - \sigma).$$

  - Recall that

$$G_\alpha = \begin{bmatrix} \sigma & 0 & \alpha \\ \alpha cnl & 1 - \alpha ns & 0 \\ c(\tau + \alpha nl) & \alpha d_1 ln & \sigma + \alpha c \end{bmatrix}, \qquad G_0 = \begin{bmatrix} \sigma & 0 & 0 \\ 0 & 1 & 0 \\ c\tau & 0 & \sigma \end{bmatrix}.$$

  - Hence, $\rho(G_0) = 1$ because $\sigma < 1$.

Convergence Rate

- **Theorem 2.** *With the step-size, $\alpha \in (0, \alpha_1)$, the sequence, $\{\mathbf{x}_k\}$, converges linearly to the optimal solution, $\mathbf{x}^*$, i.e., there exist some constant $M > 0$ such that*

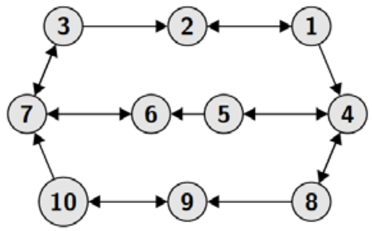
$$\|\mathbf{x}_k - \mathbf{x}^*\|_2 \leq M(\gamma + \xi)^k, \qquad \forall k,$$

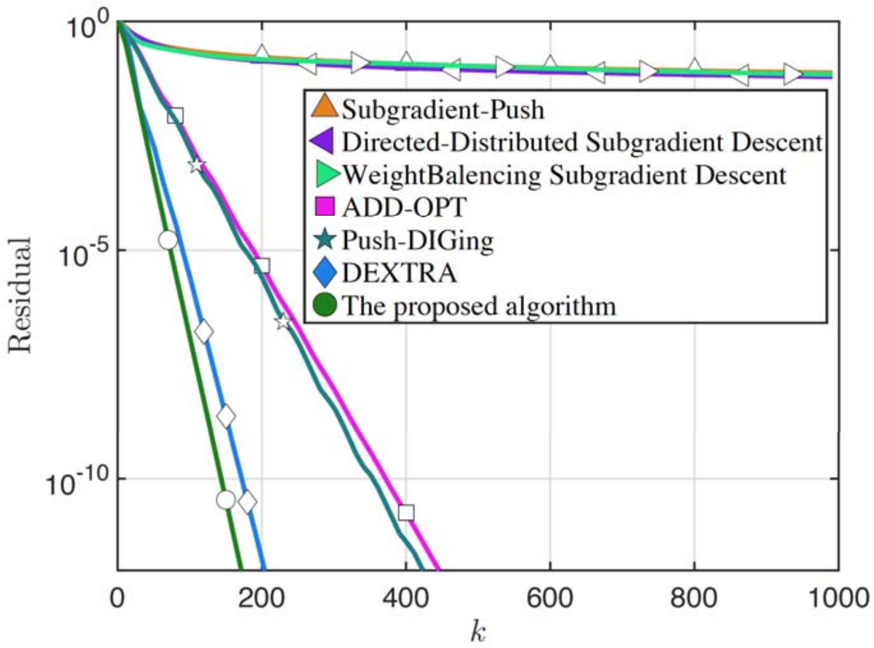
where ξ is an arbitrarily small constant.

- **The rate variable γ is the max of fusion rate and the rate at which G decays**

Some comparison

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





Conclusions

- **Optimization with row-stochastic matrices**
- **Does not require the knowledge of out-neighbors or out-degree**
  - **Agents require unique identifiers**

- **Strongly-convex functions with Lipschitz-continuous graidents**
- **Strongly-connected directed graphs**

- **Linear convergence**

More Information

- **My webpage: http://www.eecs.tufts.edu/~khan/**

- **My email: khan@ece.tufts.edu**

- **My Lab's YouTube channel: https://www.youtube.com/user/SPARTNatTufts/videos/**