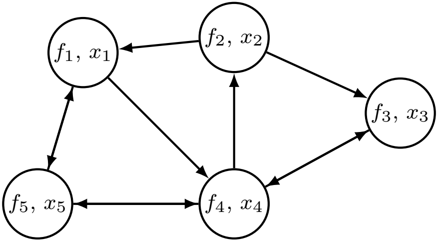
Bryan Van Scoy

bvanscoy@miamioh.edu vanscoy.github.io

Miami University

Distributed optimization

$$\begin{aligned} \underset{x_1,\ldots,x_n}{\text{minimize}} \quad & \sum_{i=1}^{n} f_i(x_i) \\ \text{subject to} \quad & x_1 = x_2 = \ldots = x_n \end{aligned}$$



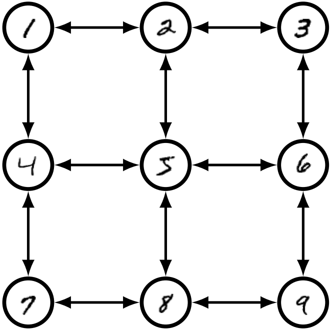
$$W = \begin{bmatrix} \frac{1}{3} & \frac{1}{3} & 0 & 0 & \frac{1}{3} \\ 0 & \frac{1}{2} & 0 & \frac{1}{2} & 0 \\ 0 & \frac{1}{6} & \frac{2}{3} & \frac{1}{6} & 0 \\ \frac{1}{3} & 0 & \frac{1}{3} & 0 & \frac{1}{3} \\ \frac{1}{3} & 0 & 0 & \frac{1}{3} & \frac{1}{3} \end{bmatrix}$$

Want each agent to compute the global optimizer $x_\star$ by communicating with local neighbors and performing local computations.

Application: Distributed machine learning

Each agent i has

- data (x_i, y_i)
- features $\Phi(x_i)$
- prediction model $\theta_i^{\mathsf{T}}\Phi(\cdot)$
- error $\|y_i - \theta_i^{\mathsf{T}}\Phi(x_i)\|$



$$\begin{array}{ll} \underset{\theta_1,\ldots,\theta_n}{\text{minimize}} & \displaystyle\sum_{i=1}^{n} \|y_i - \theta_i^{\mathsf{T}}\Phi(x_i)\| \\ \text{subject to} & \theta_1 = \theta_2 = \ldots = \theta_n \end{array}$$

MNIST dataset: `http://yann.lecun.com/exdb/mnist/`

Distributed gradient descent (DGD)

$$x_i^+ = \sum_{j=1}^{n} w_{ij}\, x_j - \alpha\, \nabla f_i\big(x_i\big)$$

- linear convergence to *suboptimal* solution with constant stepsize
- *sublinear* convergence to optimal solution with decaying stepsize

The optimal solution is *not* a fixed point.

(Nedić and Ozdaglar, 2009)

Distributed inexact gradient tracking (DIGing)

$$x_i^+ = \sum_{j=1}^{n} w_{ij}\, x_j - \alpha\, y_i$$

$$y_i^+ = \sum_{j=1}^{n} w_{ij}\, y_j + \nabla f_i(x_i^+) - \nabla f_i(x_i)$$

- linear convergence over time-varying networks
- the bound grows with the number of agents
- the bound does not apply to large stepsizes

Have conservative bounds that are specific to DIGing.

(Nedić, Olshevsky, and Shi, 2017) and (Qu, Li, 2018)

Other distributed algorithms

$$x^{+} = Wx - \alpha \nabla f(x)$$ **DGD'09**

$$x^{++} = (I + W)\, x^{+} - \tfrac{I+W}{2}\, x - \alpha \left(\nabla f(x^{+}) - \nabla f(x)\right)$$ **EXTRA'15**

$$x^{++} = W \left(2x^{+} - Wx - \alpha\, W \left(\nabla f(x^{+}) - \nabla f(x)\right)\right)$$ **AugDGM'15**

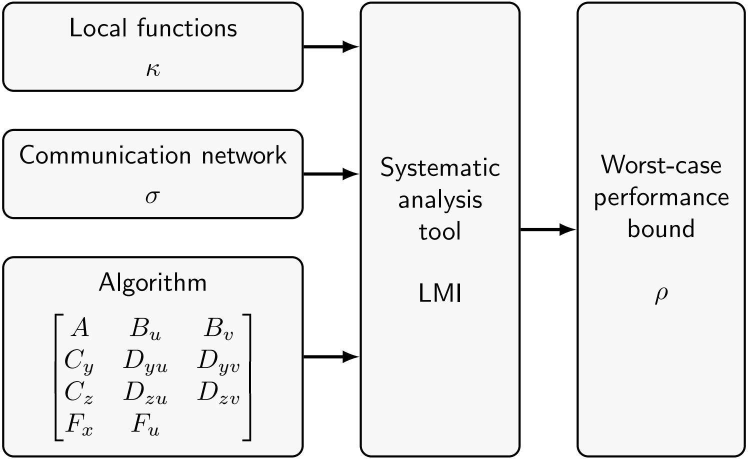
$$x^{++} = W \left(2x^{+} - Wx\right) - \alpha \left(\nabla f(x^{+}) - \nabla f(x)\right)$$ **DIGing'17**

$$x^{++} = (I + W)\, x^{+} - \tfrac{I+W}{2}\, x - \alpha \left(\nabla f\left(\tfrac{I+W}{2} x^{+}\right) - \nabla f\left(\tfrac{I+W}{2} x\right)\right)$$ **ExDiff'17**

$$x^{++} = (I + W)\, x^{+} - \tfrac{I+W}{2} \left(x + \alpha \left(\nabla f(x^{+}) - \nabla f(x)\right)\right)$$ **NIDS'19**

and more. . .

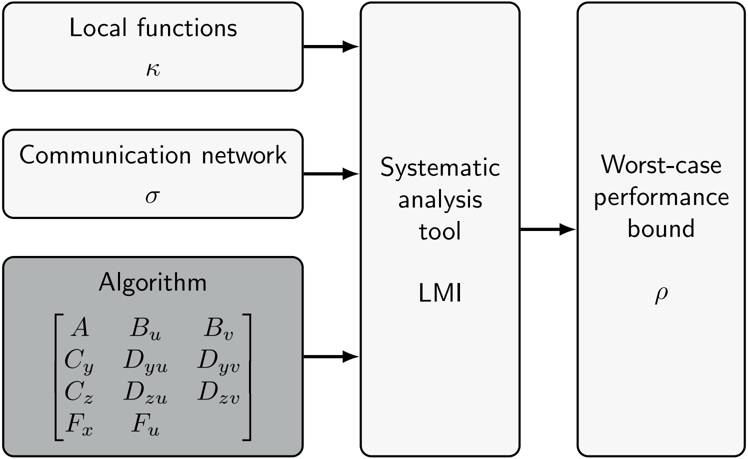
A systematic approach

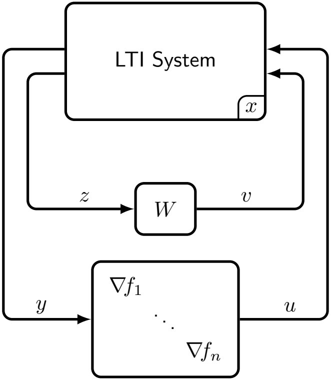


Main idea

$$V(x) \quad = \quad \underbrace{\left\| \mathsf{avg}(x) - x_\star \right\|_P^2}_{\text{optimality}} \quad + \quad \underbrace{\left\| x - \mathsf{avg}(x) \right\|_Q^2}_{\text{consensus}}$$

- search for a quadratic Lyapunov function using LMIs
- use pointwise quadratic constraints from the assumptions on the local objective functions and the communication network
- similar to Lur'e problem from robust control


Local functions
κ
Communication network
σ
Algorithm
$\begin{bmatrix} A & B_u & B_v \\ C_y & D_{yu} & D_{yv} \\ C_z & D_{zu} & D_{zv} \\ F_x & F_u & \end{bmatrix}$
Systematic analysis tool
LMI
Worst-case performance bound
ρ



$$\begin{bmatrix} x_i^+ \\ y_i \\ z_i \end{bmatrix} = \begin{bmatrix} A & B_u & B_v \\ C_y & D_{yu} & D_{yv} \\ C_z & D_{zu} & D_{zv} \end{bmatrix} \begin{bmatrix} x_i \\ u_i \\ v_i \end{bmatrix}$$

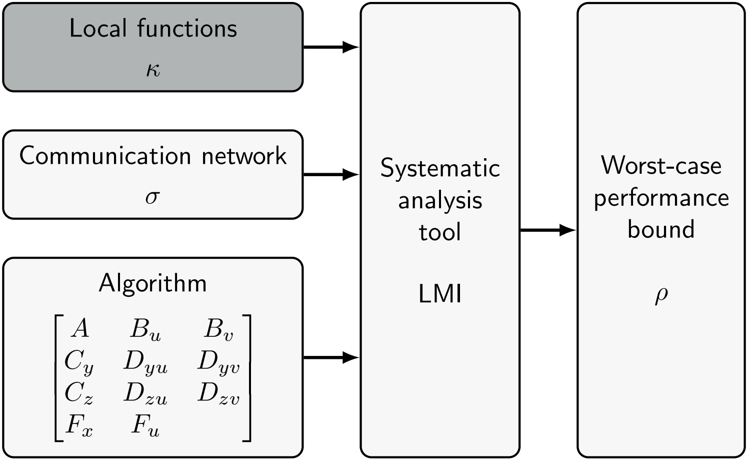
$$v_i = \sum_{j=1}^{n} w_{ij}\, z_j$$

$$u_i = \nabla f_i(y_i)$$

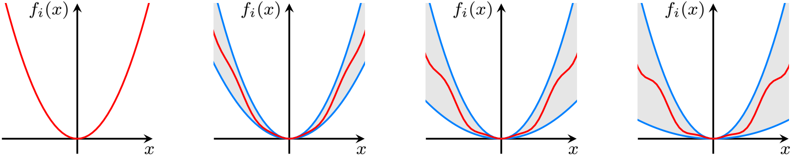
$$0 = \sum_{j=1}^{n} (F_x\, x_j + F_u\, u_j)$$

Many known algorithms fit in this form.

	2 state variables		3 state variables	
1 communicated variable	SVL	$\left[\begin{array}{cc\|c\|c} 1-\gamma & \beta & -\alpha & -\gamma \\ -1 & 1 & 0 & -1 \\ \hline 1-\delta & 0 & 0 & \delta \\ \hline 1 & 0 & 0 & 0 \\ \hline 0 & 1 & 0 & \end{array}\right]$	EXTRA	$\left[\begin{array}{ccc\|c\|c} 1 & -\frac{1}{2} & \alpha & -\alpha & 1 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ \hline 1 & 0 & 0 & 0 & 0 \\ \hline 1 & -\frac{1}{2} & 0 & 0 & 0 \\ \hline 1 & -1 & \alpha & 0 & \end{array}\right]$
	Exact Diffusion (ExDIFF)	$\left[\begin{array}{cc\|c\|c} 1 & -1 & -\alpha & 1 \\ \frac{1}{2} & 0 & -\alpha & \frac{1}{2} \\ \hline 1 & 0 & -\frac{1}{2} & 0 \\ \hline 1 & 0 & 0 & 0 \\ \hline 1 & -1 & 0 & \end{array}\right]$	NIDS	$\left[\begin{array}{ccc\|c\|c} 1 & -\frac{1}{2} & \frac{\alpha}{2} & -\frac{\alpha}{2} & 1 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ \hline 1 & 0 & 0 & 0 & 0 \\ \hline 1 & -\frac{1}{2} & \frac{\alpha}{2} & -\frac{\alpha}{2} & 0 \\ \hline 1 & -1 & \alpha & 0 & \end{array}\right]$
2 communicated variables	Unified DIGing (uDIG)	$\left[\begin{array}{cc\|c\|cc} 0 & -\alpha & -\alpha & 1 & 0 \\ \frac{L+m}{2} & 0 & -1 & 0 & 1 \\ \hline 1 & 0 & 0 & 0 & 0 \\ \hline 1 & 0 & 0 & 0 & 0 \\ -\frac{L+m}{2} & 1 & 1 & 0 & 0 \\ \hline 0 & 1 & 0 & & \end{array}\right]$	DIGing	$\left[\begin{array}{ccc\|c\|cc} 0 & -\alpha & 0 & 0 & 1 & 0 \\ 0 & 0 & -1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ \hline 0 & -\alpha & 0 & 0 & 1 & 0 \\ \hline 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ \hline 0 & 1 & -1 & 0 & & \end{array}\right]$
	Unified EXTRA (uEXTRA)	$\left[\begin{array}{cc\|c\|cc} 0 & -\alpha & -\alpha & 1 & 0 \\ 0 & 0 & -1 & L & 1 \\ \hline 1 & 0 & 0 & 0 & 0 \\ \hline 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & -L & 0 \\ \hline 0 & 1 & 0 & & \end{array}\right]$	AugDGM	$\left[\begin{array}{ccc\|c\|cc} 0 & 0 & 0 & 0 & 1 & -\alpha \\ 0 & 0 & -1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & 1 & -\alpha \\ \hline 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ \hline 0 & 1 & -1 & 0 & & \end{array}\right]$


Local functions
κ
Communication network
σ
Algorithm
$\begin{bmatrix} A & B_u & B_v \\ C_y & D_{yu} & D_{yv} \\ C_z & D_{zu} & D_{zv} \\ F_x & F_u & \end{bmatrix}$
Systematic analysis tool
LMI
Worst-case performance bound
ρ

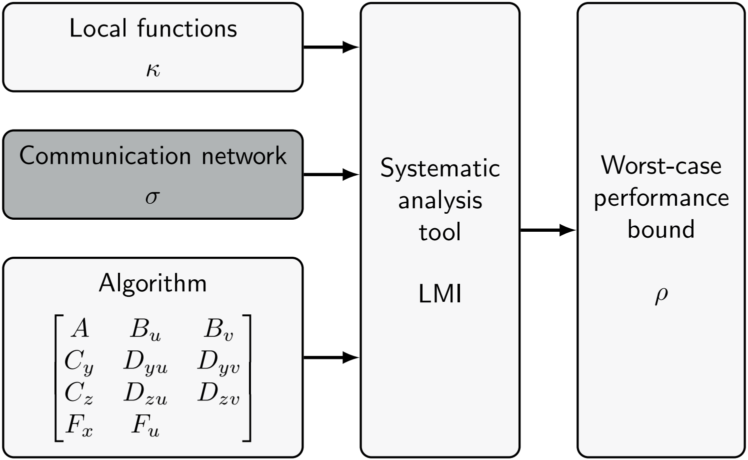
Each local objective function f_i is L-smooth and m-strongly convex with respect to the global optimizer $x_\star$.



$\kappa = 1$ $\kappa = 2$ $\kappa = 4$ $\kappa = 8$

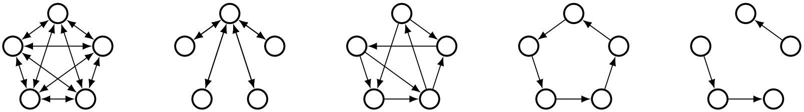
The condition ratio $\kappa = L/m$ characterizes the variation in curvature.

$$\begin{bmatrix} \nabla f_i(y) - \nabla f_i(y_\star) \\ y - y_\star \end{bmatrix}^\mathsf{T} \begin{bmatrix} -2mL & L+m \\ L+m & -2 \end{bmatrix} \begin{bmatrix} \nabla f_i(y) - \nabla f_i(y_\star) \\ y - y_\star \end{bmatrix} \geq 0$$



Local functions
κ
Communication network
σ
Algorithm
$\begin{bmatrix} A & B_u & B_v \\ C_y & D_{yu} & D_{yv} \\ C_z & D_{zu} & D_{zv} \\ F_x & F_u & \end{bmatrix}$
Systematic analysis tool
LMI
Worst-case performance bound
ρ

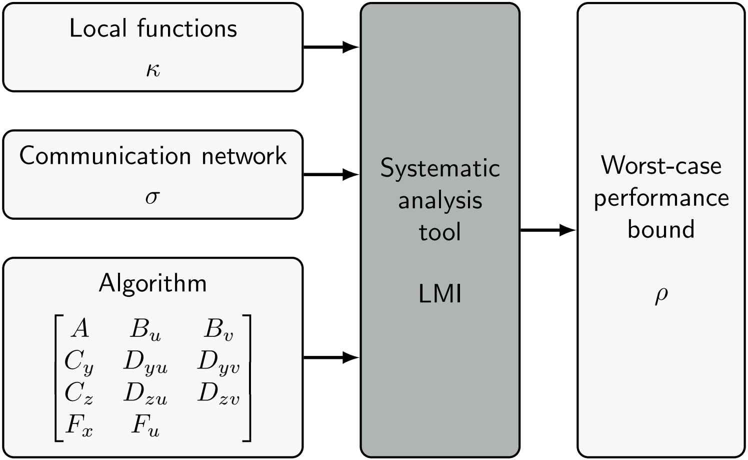
At each iteration, the gossip matrix W satisfies the sparsity pattern of the graph, is doubly stochastic, and has spectral gap σ.



$\sigma = 0.0$ $\quad$ $\sigma = 0.667$ $\quad$ $\sigma = 0.707$ $\quad$ $\sigma = 0.809$ $\quad$ $\sigma = 1.0$

The spectral gap $\sigma = \left\| \frac{1}{n}\mathbf{1}\mathbf{1}^\mathsf{T} - W \right\|_2$ characterizes network connectivity.

$$\begin{bmatrix} z - \mathsf{avg}(z) \\ Wz - \mathsf{avg}(z) \end{bmatrix}^\mathsf{T} \begin{bmatrix} \sigma^2 & 0 \\ 0 & -1 \end{bmatrix} \begin{bmatrix} z - \mathsf{avg}(z) \\ Wz - \mathsf{avg}(z) \end{bmatrix} \geq 0$$


Local functions
κ
Communication network
σ
Algorithm
$\begin{bmatrix} A & B_u & B_v \\ C_y & D_{yu} & D_{yv} \\ C_z & D_{zu} & D_{zv} \\ F_x & F_u & \end{bmatrix}$
Systematic analysis tool
LMI
Worst-case performance bound
ρ

Optimality:

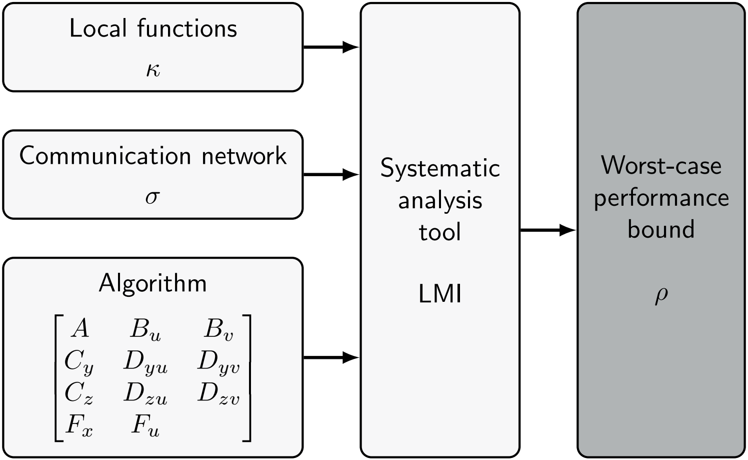
$$\Psi^\mathsf{T} \left[\begin{array}{cc} A & B_u \\ I & 0 \\ \hdashline C_y & D_{yu} \\ 0 & I \end{array}\right]^\mathsf{T} \left[\begin{array}{cc:c} P & 0 & 0 \\ 0 & -\rho^2 P & 0 \\ \hdashline 0 & 0 & M_0 \end{array}\right] \left[\begin{array}{cc} A & B_u \\ I & 0 \\ \hdashline C_y & D_{yu} \\ 0 & I \end{array}\right] \Psi \preceq 0$$

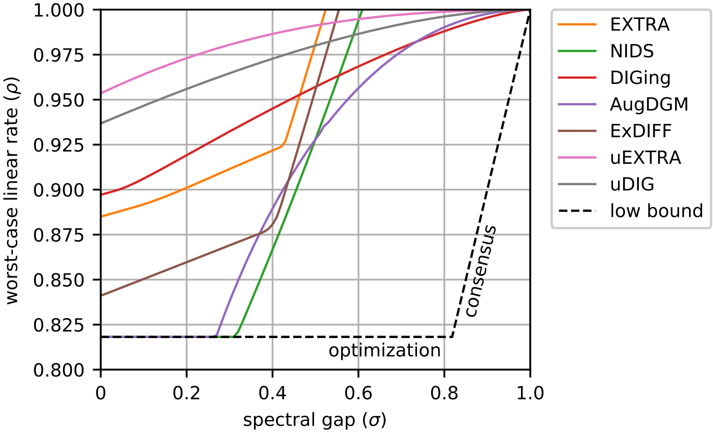
Consensus:

$$\left[\begin{array}{ccc} A & B_u & B_v \\ I & 0 & 0 \\ \hdashline C_y & D_{yu} & D_{yv} \\ 0 & I & 0 \\ \hdashline C_z & D_{zu} & D_{zv} \\ 0 & 0 & I \end{array}\right]^\mathsf{T} \left[\begin{array}{cc:c:c} Q & 0 & 0 & 0 \\ 0 & -\rho^2 Q & 0 & 0 \\ \hdashline 0 & 0 & M_0 & 0 \\ \hdashline 0 & 0 & 0 & M_1 \otimes R \end{array}\right] \left[\begin{array}{ccc} A & B_u & B_v \\ I & 0 & 0 \\ \hdashline C_y & D_{yu} & D_{yv} \\ 0 & I & 0 \\ \hdashline C_z & D_{zu} & D_{zv} \\ 0 & 0 & I \end{array}\right] \preceq 0$$

$$M_0 = \begin{bmatrix} -2\kappa & \kappa+1 \\ \kappa+1 & -2 \end{bmatrix} \qquad M_1 = \begin{bmatrix} \sigma^2 - 1 & 1 \\ 1 & -1 \end{bmatrix} \qquad \Psi = \text{null}\begin{bmatrix} F_x & F_u \end{bmatrix}$$

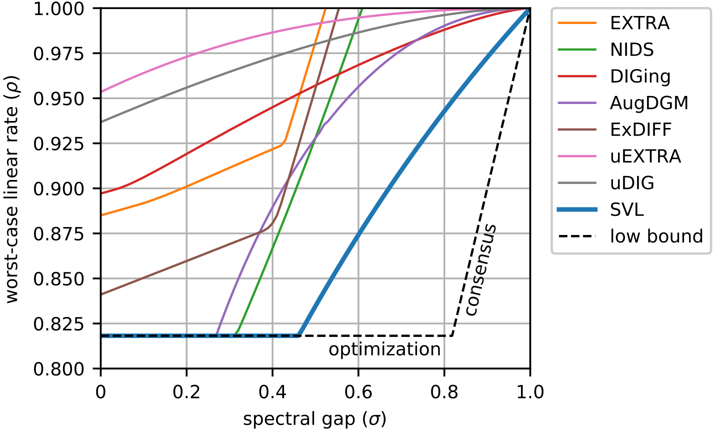
Feasibility implies each agent converges to $x_\star$ linearly with rate ρ.


Local functions
κ
Communication network
σ
Algorithm
$\begin{bmatrix} A & B_u & B_v \\ C_y & D_{yu} & D_{yv} \\ C_z & D_{zu} & D_{zv} \\ F_x & F_u & \end{bmatrix}$
Systematic analysis tool
LMI
Worst-case performance bound
ρ



- $\kappa = 10$ and the gradient stepsize α is optimized
- lower bound corresponds to only optimization and only consensus

Can analyze *all* algorithms using the same technique.



- SVL has the best worst-case convergence
- equivalent to inexact ADMM **(Boyd et al., 2011)**

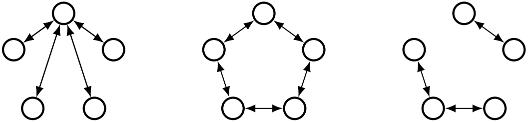
The analysis LMIs can also be used for *design*.

Extensions

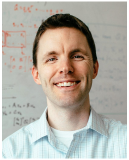
- optimized trade-off between communication and computation

iteration	1	2	3	4	5	6	7	8	9
communication	✓	✓	✓	✓	✓	✓	✓	✓	✓
gradient evaluation			✓			✓			✓

- jointly-connected communication networks



the network is connected only when averaged over time



Laurent Lessard

Akhil Sundararajan

TCNS'20 A. Sundararajan, B. Van Scoy, and L. Lessard, "Analysis and design of first-order distributed optimization algorithms over time-varying graphs"

CDC'20 B. Van Scoy and L. Lessard, "Systematic analysis of distributed optimization algorithms over jointly-connected networks"

ACC'19 A. Sundararajan, B. Van Scoy, and L. Lessard, "A canonical form for first-order distributed optimization algorithms"

NecSys'19 B. Van Scoy and L. Lessard, "A distributed optimization algorithm over time-varying graphs with efficient gradient evaluations"

`https://vanscoy.github.io`