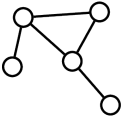
Multi-Agent Systems: Systematic Analysis and Design

Bryan Van Scoy

Miami University

Multi-agent systems



A system composed of multiple interacting agents.

drone swarms

vehicle platoons

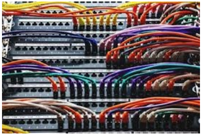
smart grid

wind farm

smart factory

server farm

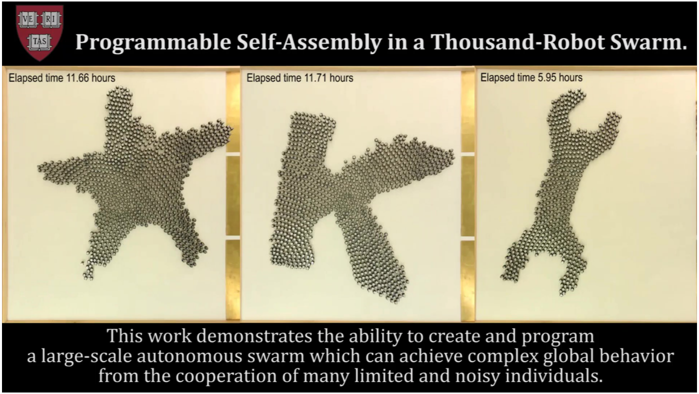
sports



routing

Kilobot: Mike Rubenstein (2014)

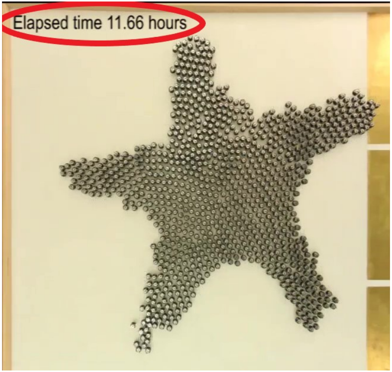
- 1,024 robot swarm
- 3.3cm tall
- total cost under $15 per robot



Olympic Drone Show: Intel (2018)

- 1,218 drone swarm
- about a foot long
- can fly in formation for 20 minutes

Issues with current implementations



inefficient

centralized and open-loop

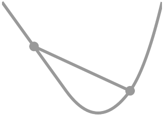
Problem challenges

- large-scale
- robust
- efficient

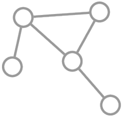
smart grid

Optimization + Robust control

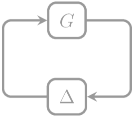
Optimization



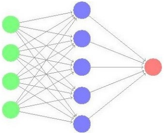
Multi-agent systems



Robust control



Making the best or most effective use of a situation or resource.An approach to controller design that explicitly deals with uncertainty.



machine learning

navigation

resource allocation

portfolio

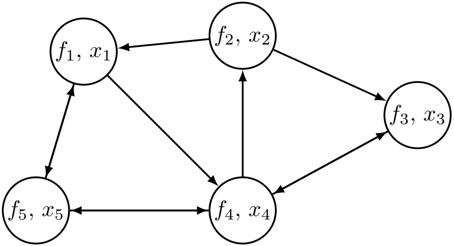
drones

robotics

aircraft

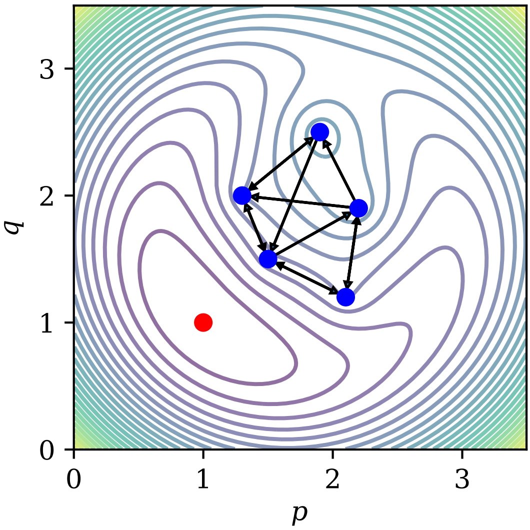
Problem abstraction

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



Want each agent to compute the global optimizer x_{opt} by communicating with local neighbors and performing local computations.

Example: Target localization

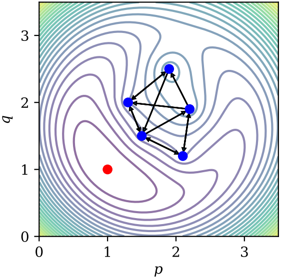


Example: Target localization

- target located at position $(p_\star, q_\star) \in \mathbb{R}^2$

- agent i knows its position $(p_i, q_i) \in \mathbb{R}^2$ and distance to the target

$$r_i = \sqrt{(p_i - p_\star)^2 + (q_i - q_\star)^2}$$



- the objective function associated to agent i is

$$f_i(p, q) = \frac{1}{2}\left(\sqrt{(p_i - p)^2 + (q_i - q)^2} - r_i\right)^2$$

- the target position is the solution to the problem

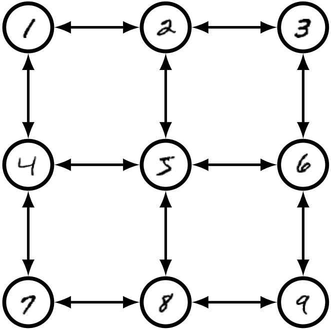
$$\underset{p,\,q}{\text{minimize}} \quad \sum_{i=1}^{n} f_i(p, q)$$

Example: Distributed machine learning

Each agent i has

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

$$\underset{\theta}{\text{minimize}} \quad \sum_{i=1}^{n} \|y_i - \theta^{\mathsf{T}}\Phi(x_i)\|$$

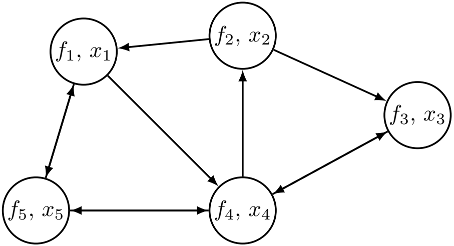


Agents want a *global* model while keeping their local data *private*.

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

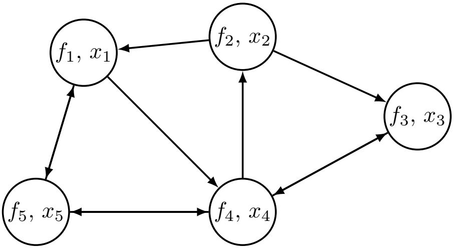
Problem abstraction

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



Want each agent to compute the global optimizer x_{opt} by communicating with local neighbors and performing local computations.

Gossip matrix



- $w_{ij} = 0$ if agent i does not receive information from agent j
- doubly stochastic

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

Distributed gradient descent (Nedić & Ozdaglar, 2009)

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

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

The optimal solution is *not* a fixed point...

Other distributed algorithms

$$x^{k+1} = Wx^k - \alpha \nabla f(x^k)$$ **DGD'09**

$$x^{k+1} = (I+W)x^k - \tfrac{I+W}{2}x^{k-1} - \alpha \nabla f(x^k) + \alpha \nabla f(x^{k-1})$$ **EXTRA'15**

$$x^{k+1} = 2Wx^k - W^2 x^{k-1} - \alpha W^2 \big(\nabla f(x^k) - \nabla f(x^{k-1})\big)$$ **AugDGM'15**

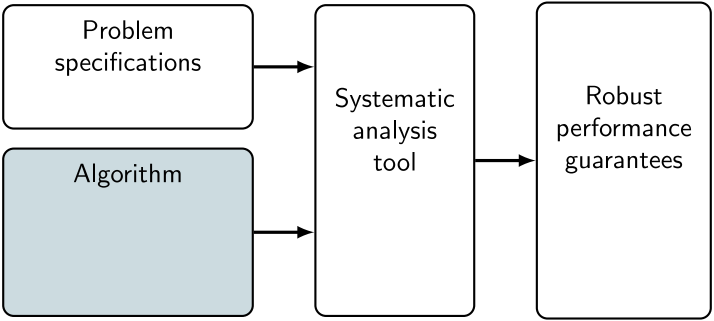
$$x^{k+1} = 2Wx^k - W^2 x^{k-1} - \alpha \nabla f(x^k) + \alpha \nabla f(x^{k-1})$$ **DIGing'17**

$$x^{k+1} = (I+W)x^k - \tfrac{I+W}{2}x^{k-1} - \alpha \nabla f(\tfrac{I+W}{2}x^k) + \alpha \nabla f(\tfrac{I+W}{2}x^{k+1})$$ **ExDiff'17**

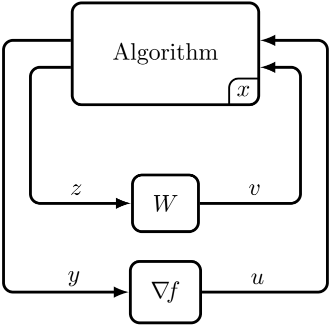
$$x^{k+1} = (I+W)x^k - \tfrac{I+W}{2}\big(x^{k-1} + \alpha \nabla f(x^k) - \alpha \nabla f(x^{k-1})\big)$$ **NIDS'19**

and more. . .

A systematic approach



General algorithm form



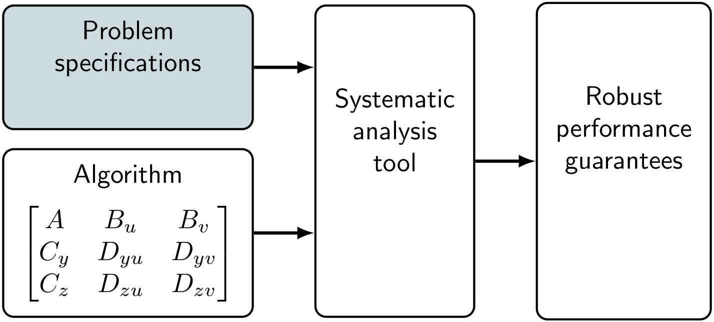
$$\begin{bmatrix} x_i^{k+1} \\ y_i^k \\ z_i^k \end{bmatrix} = \begin{bmatrix} A & B_u & B_v \\ C_y & D_{yu} & D_{yv} \\ C_z & D_{zu} & D_{zv} \end{bmatrix} \begin{bmatrix} x_i^k \\ u_i^k \\ v_i^k \end{bmatrix}$$

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

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

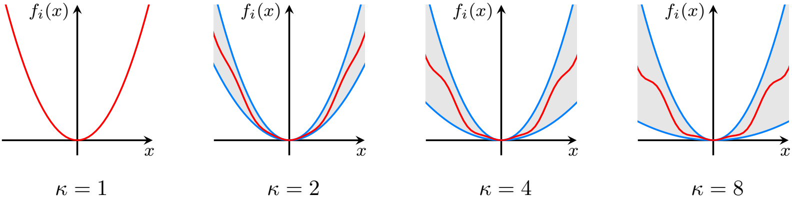
Many known algorithms fit in this form!

	2 state variables	3 state variables
1 communicated variable	**Template:** $\begin{bmatrix} A & B_u & B_v \\ C_y & D_{yu} & D_{yv} \\ C_z & D_{zu} & D_{zv} \end{bmatrix}$	EXTRA $\left[\begin{array}{ccc\|c\|c} 1 & -\frac{1}{2} & \alpha & -\alpha & 1 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ \hline 1 & 0 & 0 & 0 & 0 \\ \hline 1 & -\frac{1}{2} & 0 & 0 & 0 \end{array}\right]$
	Exact Diffusion (ExDIFF) $\left[\begin{array}{cc\|c\|c} 1 & -1 & -\alpha & 1 \\ \frac{1}{2} & 0 & -\alpha & \frac{1}{2} \\ \hline 1 & 0 & -\frac{1}{2} & 0 \\ \hline 1 & 0 & 0 & 0 \end{array}\right]$	NIDS $\left[\begin{array}{ccc\|c\|c} 1 & -\frac{1}{2} & \frac{\alpha}{2} & -\frac{\alpha}{2} & 1 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ \hline 1 & 0 & 0 & 0 & 0 \\ \hline 1 & -\frac{1}{2} & \frac{\alpha}{2} & -\frac{\alpha}{2} & 0 \end{array}\right]$
2 communicated variables	Unified DIGing (uDIG) $\left[\begin{array}{cc\|c\|cc} 0 & -\alpha & -\alpha & 1 & 0 \\ \frac{L+m}{2} & 0 & -1 & 0 & 1 \\ \hline 1 & 0 & 0 & 0 & 0 \\ \hline 1 & 0 & 0 & 0 & 0 \\ -\frac{L+m}{2} & 1 & 1 & 0 & 0 \end{array}\right]$	DIGing $\left[\begin{array}{ccc\|c\|cc} 0 & -\alpha & 0 & 0 & 1 & 0 \\ 0 & 0 & -1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ \hline 0 & -\alpha & 0 & 0 & 1 & 0 \\ \hline 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \end{array}\right]$
	Unified EXTRA (uEXTRA) $\left[\begin{array}{cc\|c\|cc} 0 & -\alpha & -\alpha & 1 & 0 \\ 0 & 0 & -1 & L & 1 \\ \hline 1 & 0 & 0 & 0 & 0 \\ \hline 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & -L & 0 \end{array}\right]$	AugDGM $\left[\begin{array}{ccc\|c\|cc} 0 & 0 & 0 & 0 & 1 & -\alpha \\ 0 & 0 & -1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & 1 & -\alpha \\ \hline 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \end{array}\right]$


Problem specifications
Algorithm
$\begin{bmatrix} A & B_u & B_v \\ C_y & D_{yu} & D_{yv} \\ C_z & D_{zu} & D_{zv} \end{bmatrix}$
Systematic analysis tool
Robust performance guarantees

Objective functions

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



The condition ratio $\kappa := L/m$ characterizes how much the curvature varies.

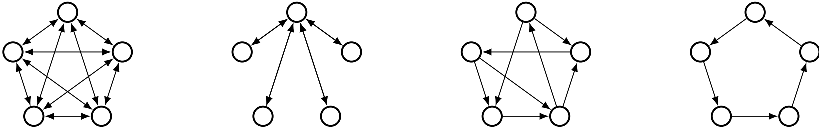
For all x,

$$\big(\nabla f_i(x) - \nabla f_i(x_{\mathsf{opt}}) - m\,(x - x_{\mathsf{opt}})\big)^{\mathsf{T}}\big(\nabla f_i(x) - \nabla f_i(x_{\mathsf{opt}}) - L\,(x - x_{\mathsf{opt}})\big) \leq 0$$

Communication graphs

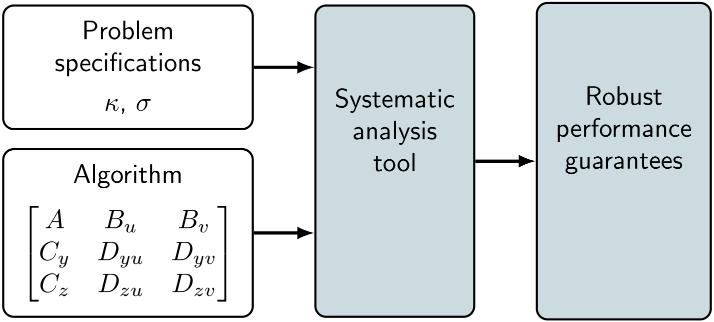
At each step of the algorithm, the following properties hold:

Graph	**Gossip matrix**
connected	eigenvalue at one has multiplicity one
balanced	doubly stochastic
bounded norm	$\left\lVert \frac{1}{n}\mathbf{1}\mathbf{1}^{\mathsf{T}} - W^k \right\rVert_2 \le \sigma$



$\sigma = 0$ $\quad$ $\sigma = 0.667$ $\quad$ $\sigma = 0.707$ $\quad$ $\sigma = 0.809$

The spectral gap σ characterizes how connected the graph is.


Problem specifications
κ, σ
Algorithm
$\begin{bmatrix} A & B_u & B_v \\ C_y & D_{yu} & D_{yv} \\ C_z & D_{zu} & D_{zv} \end{bmatrix}$
Systematic analysis tool
Robust performance guarantees

Main analysis result

Define the matrices $M_0 = \left[\begin{smallmatrix} -2mL & L+m \\ L+m & -2 \end{smallmatrix}\right]$ and $M_1 = \left[\begin{smallmatrix} \sigma^2-1 & 1 \\ 1 & -1 \end{smallmatrix}\right]$.

Suppose there exist matrices $P \succ 0$, $Q \succ 0$, and $R \succeq 0$ such that

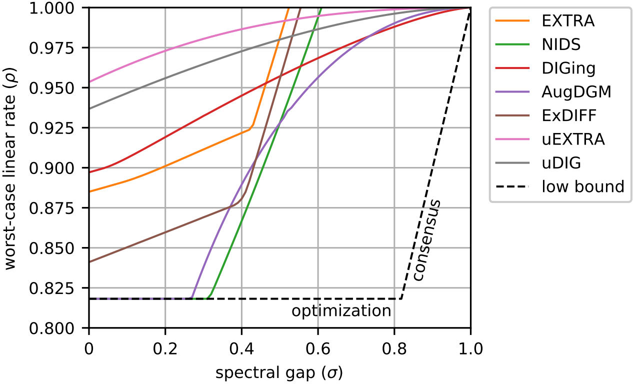
$$\begin{bmatrix} A & B_u \\ I & 0 \\ \hline C_y & D_{yu} \\ 0 & I \end{bmatrix}^\mathsf{T} \left[\begin{array}{cc|c} P & 0 & 0 \\ 0 & -\rho^2 P & 0 \\ \hline 0 & 0 & M_0 \end{array}\right] \begin{bmatrix} A & B_u \\ I & 0 \\ \hline C_y & D_{yu} \\ 0 & I \end{bmatrix} \preceq 0,$$

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

Then each agent's estimate converges to the globally optimal solution linearly with rate ρ.

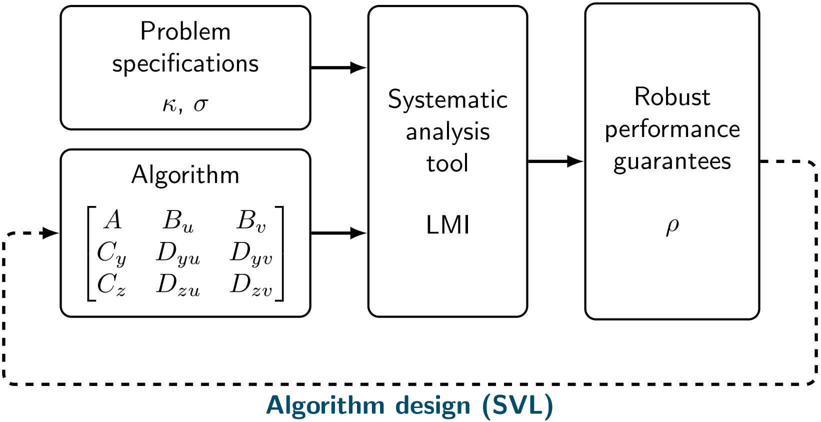
There exists $c > 0$ such that $\|y_i^k - x_{\mathsf{opt}}\| \leq c\,\rho^k$ for all agents i and all iterations k.

Algorithm comparison



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

Can analyze *all* algorithms using the same technique!


Problem specifications
κ, σ
Algorithm
$\begin{bmatrix} A & B_u & B_v \\ C_y & D_{yu} & D_{yv} \\ C_z & D_{zu} & D_{zv} \end{bmatrix}$
Systematic analysis tool
LMI
Robust performance guarantees
ρ
Algorithm design (SVL)

Algorithm design: SVL

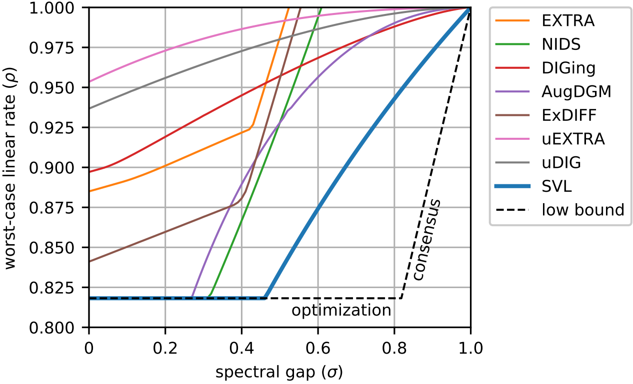
$$v_i^k = \sum_{j=1}^{n} w_{ij}^k \left(\begin{bmatrix} 1 & 0 \end{bmatrix} x_i^k \right)$$ local communication

$$u_i^k = \nabla f_i \Big(\begin{bmatrix} 1-\delta & 0 \end{bmatrix} x_i^k + \delta\, v_i^k \Big)$$ local gradient computation

$$x_i^{k+1} = \begin{bmatrix} 1-\gamma & \beta \\ -1 & 1 \end{bmatrix} x_i^k + \begin{bmatrix} -\alpha \\ 0 \end{bmatrix} u_i^k + \begin{bmatrix} \gamma \\ 1 \end{bmatrix} v_i^k$$ local state update

Given (κ, σ), choose $(\alpha, \beta, \gamma, \delta)$ to minimize ρ subject to the matrix inequalities being feasible.

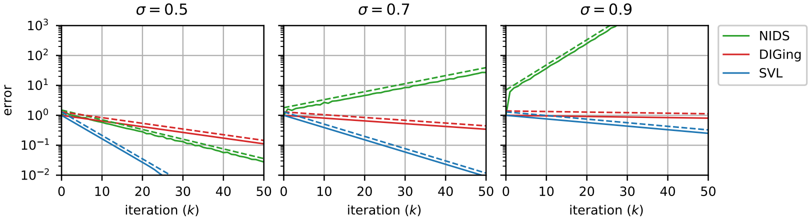
Algorithm comparison with SVL



Best worst-case performance for *any* known algorithm!

But are the bounds tight?

Worst-case trajectories



- greedily choose the functions and graph at each iteration to maximize the error
- requires solving a small (nonconvex) QCQP at each iteration

Numerical validation that the theoretical rates are tight.

A simple interpretation of SVL

ADMM

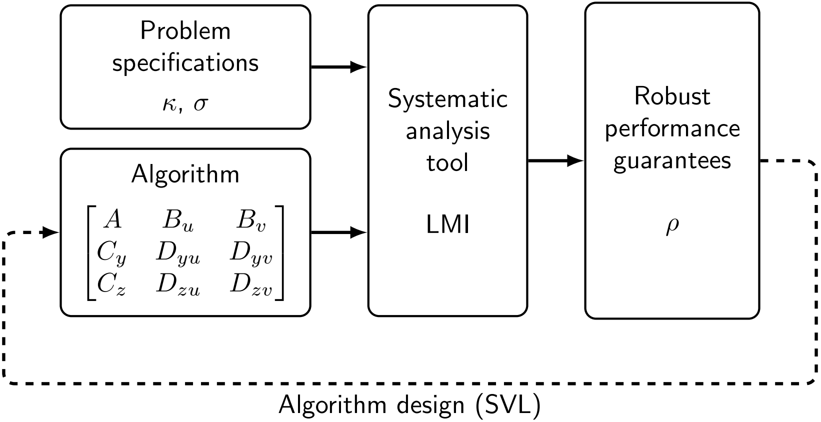
$$\begin{cases} x_i^{k+1} = \arg\min_x f_i(x) + \left(x - y_i^k\right)^\mathsf{T} z_i^k + \frac{\beta}{2}\left\|x - y_i^k\right\|^2 \\ y_i^{k+1} = \frac{1}{n}\sum_{j=1}^{n} x_j^{k+1} \\ z_i^{k+1} = z_i^k + \beta\left(x_i^{k+1} - y_i^{k+1}\right) \end{cases}$$

Inexact ADMM

$$\begin{cases} x_i^{k+1} = y_i^k - \alpha\left(\nabla f_i(y_i^k) + z_i^k\right) \\ y_i^{k+1} = \sum_{j=1}^{n} w_{ij}^{k+1}\, x_j^{k+1} \\ z_i^{k+1} = z_i^k + \beta\,(x_i^{k+1} - y_i^{k+1}) \end{cases}$$

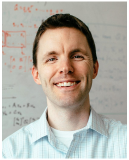
SVL is equivalent to inexact ADMM!

Summary



- developed systematic analysis of distributed optimization algorithms
- designed the fastest algorithm for the case of time-varying networks
- provided a simple interpretation of our algorithm as inexact ADMM

Collaborators



Laurent Lessard

Akhil Sundararajan

Selected References

- A. Sundararajan, B. Van Scoy, and L. Lessard, "Analysis and design of first-order distributed optimization algorithms over time-varying graphs," *IEEE Transactions on Control of Networked Systems*, 2020 (to appear).
- B. Van Scoy and L. Lessard, "A distributed optimization algorithm over time-varying graphs with efficient gradient evaluations," *NecSys*, 2019.
- A. Sundararajan, B. Van Scoy, and L. Lessard, "A canonical form for first-order distributed optimization algorithms," *ACC*, 2019.

Website

`https://vanscoy.github.io`