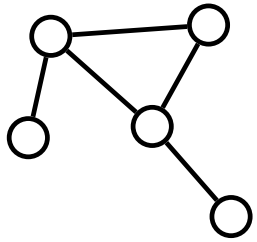
Bryan Van Scoy

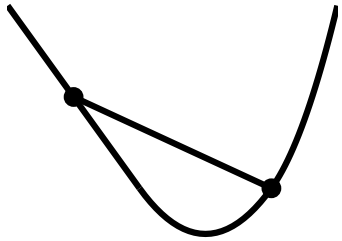
bvanscoy@miamioh.edu vanscoy.github.io

Miami University

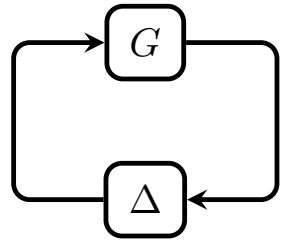
Multi-agent systems



Optimization

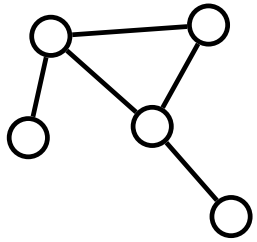


Robust control

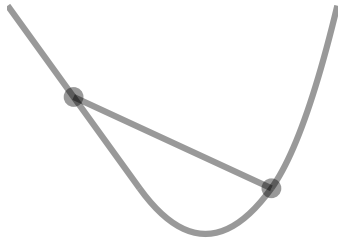


All of my teaching and research involves dynamical systems.

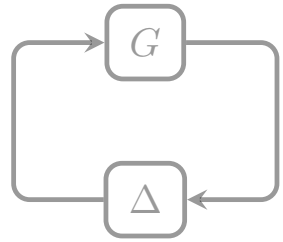
Multi-agent systems



Optimization



Robust control



A system composed of multiple interacting agents.

drone swarms

vehicle platoons

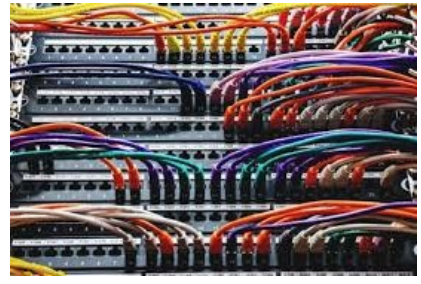
smart grid

wind farm

smart factory

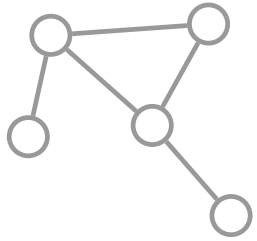
server farm

sports

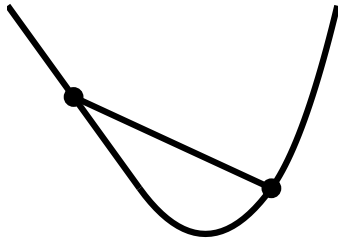


routing

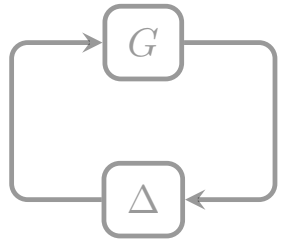
Multi-agent systems



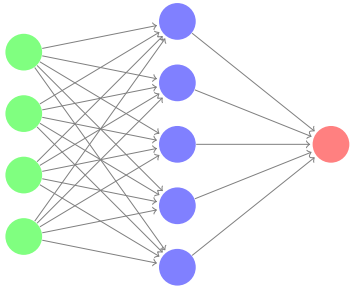
Optimization



Robust control



Making the best or most effective use of a situation or resource.



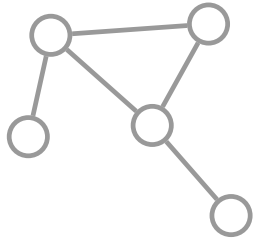
machine learning

navigation

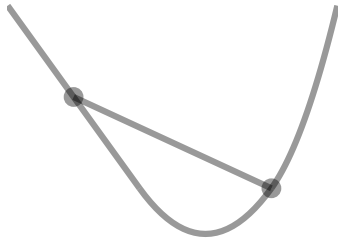
resource allocation

portfolio optimization

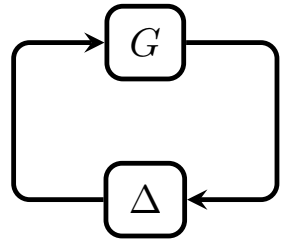
Multi-agent systems



Optimization



Robust control



An approach to controller design that explicitly deals with uncertainty.

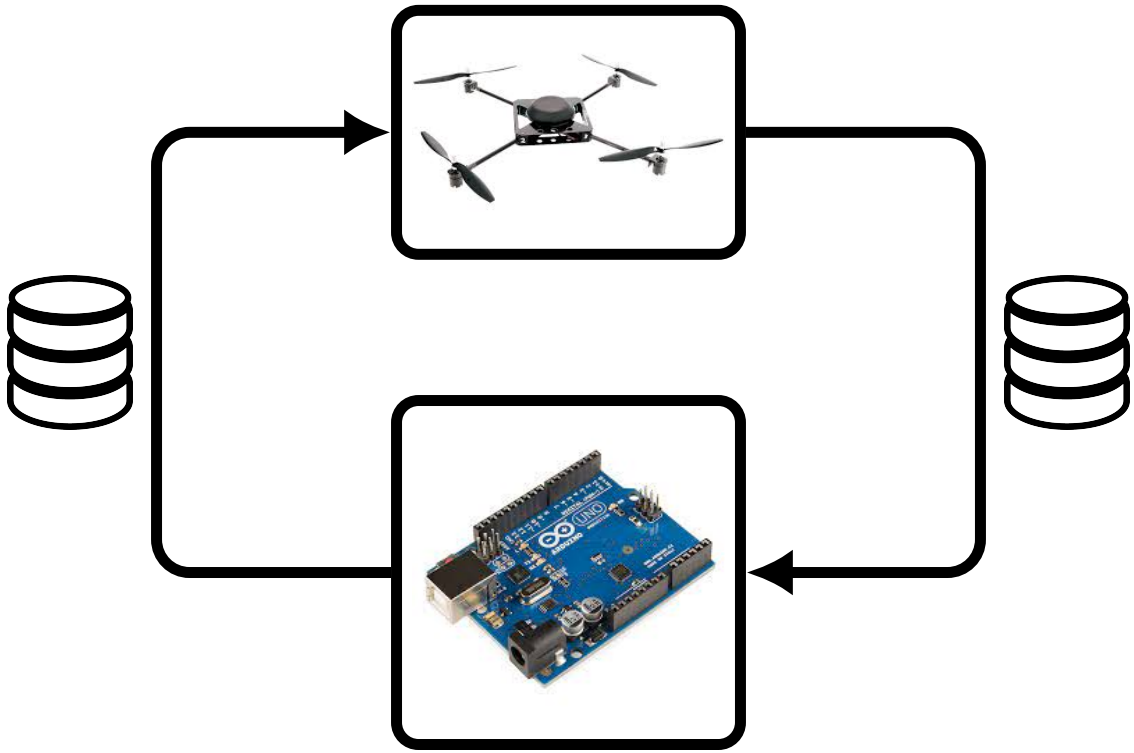
drones

robotics

aircraft

factories

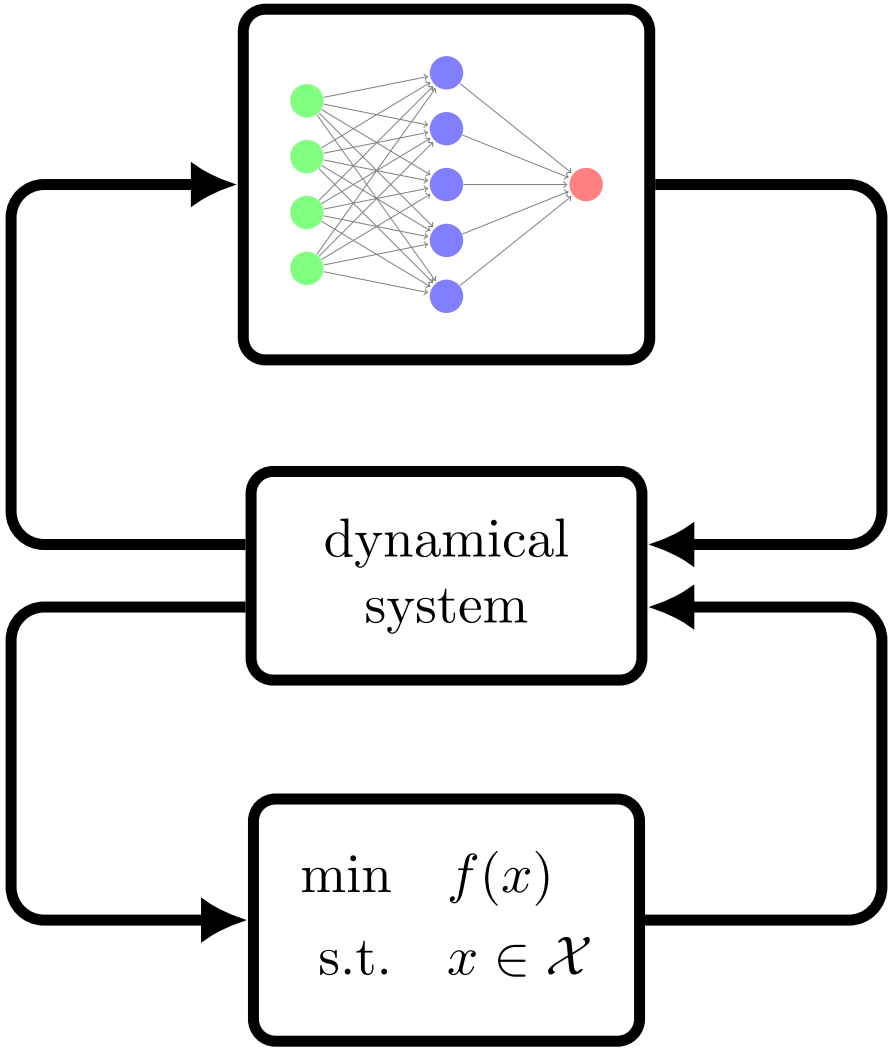
Data-driven control



- many systems are too complex for a simple model
- use data to directly control the system
- connections with *optimization* and *reinforcement learning*

quadrotor controlled by an Arduino microcontroller

Complex interconnected systems



Multi-agent robotics lab

Kilobot by Mike Rubenstein

Olympic Drone Show by Intel

- provide hands-on experience with various robotic systems
- utilize for capstone design projects
- testbed for algorithm development

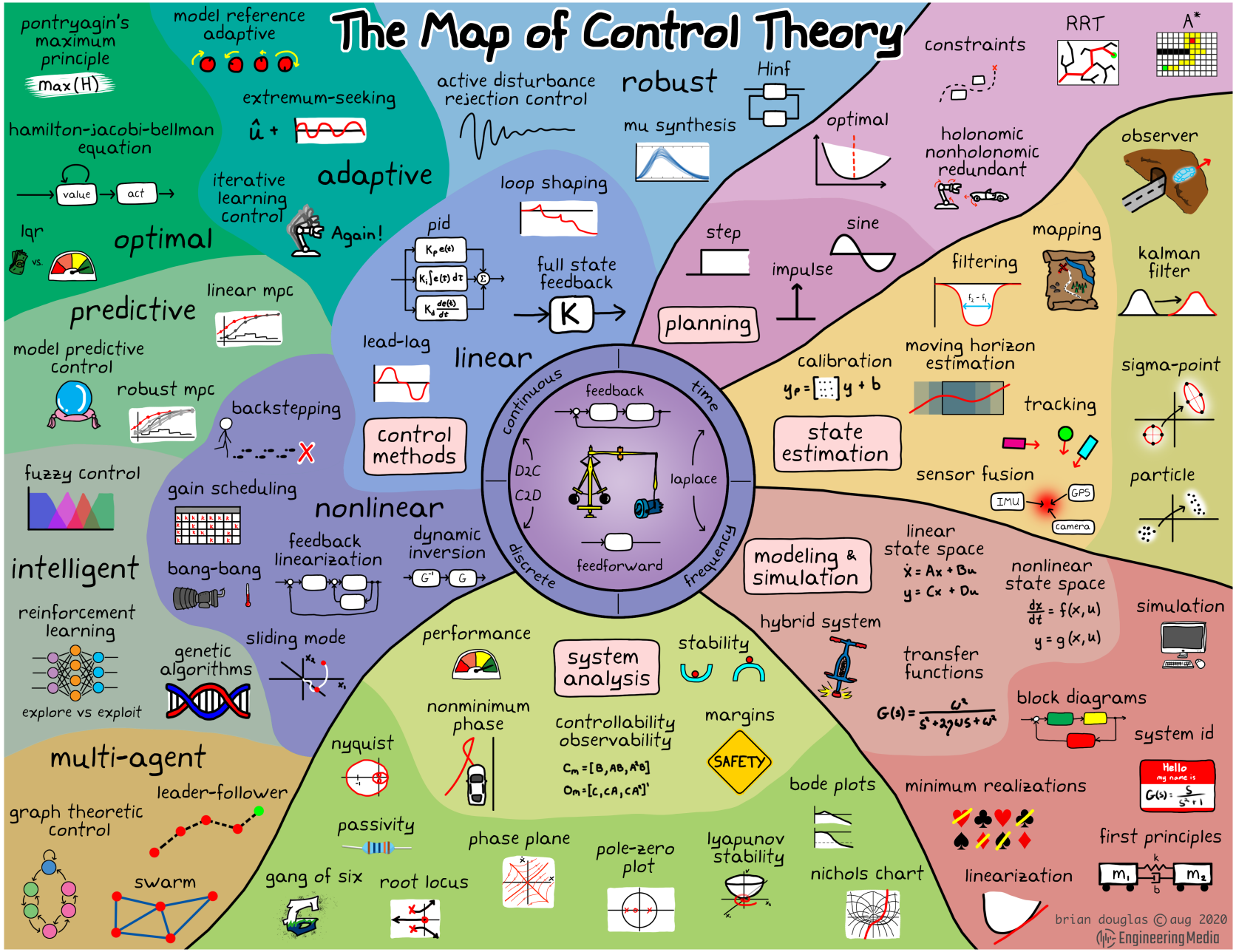
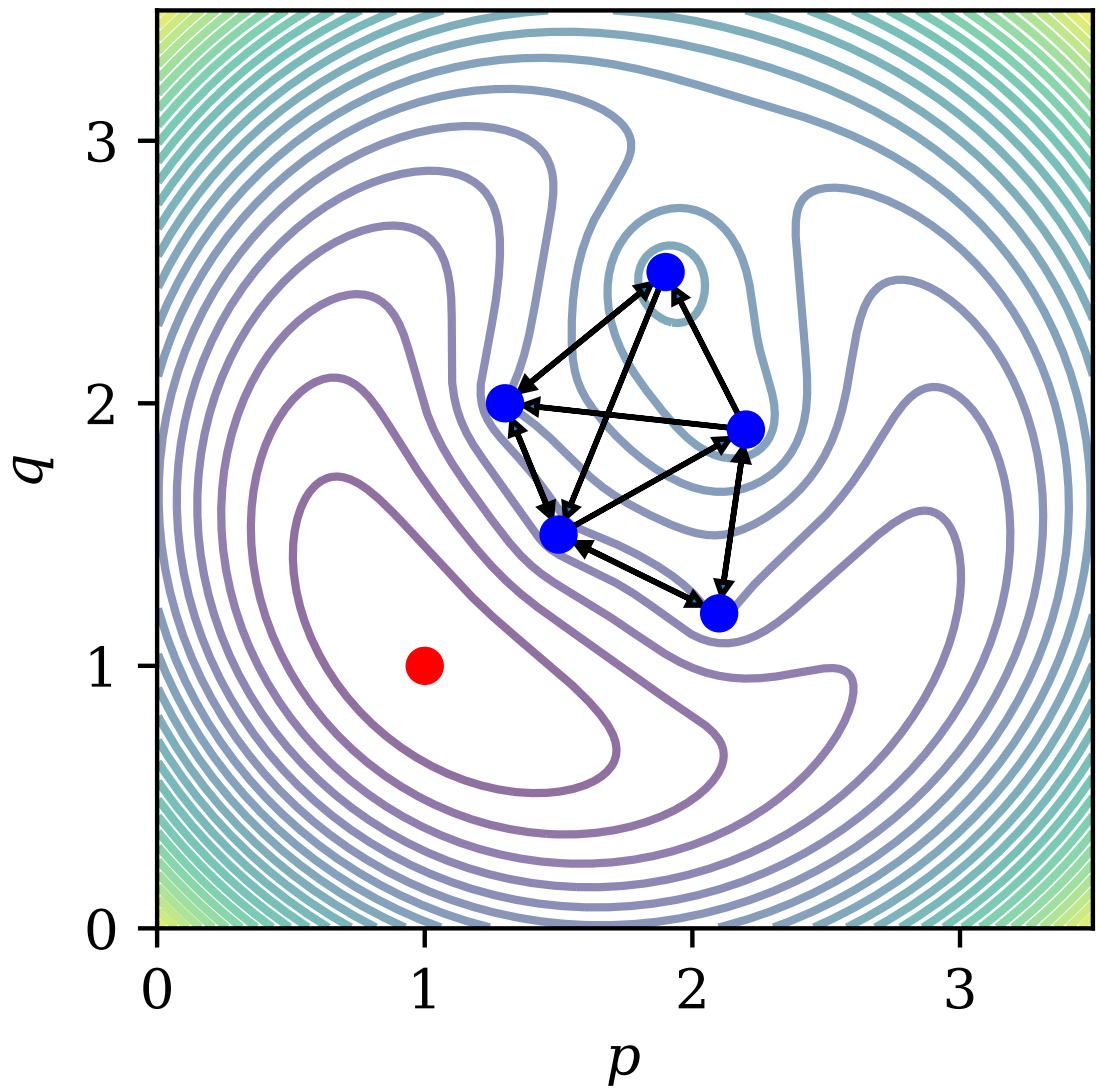


image by Brian Douglas at Engineering Media

Example: Target localization

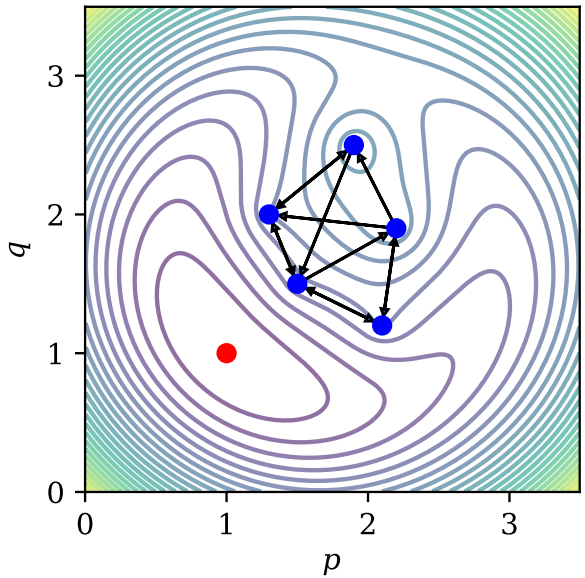


Boyd and Vandenberghe. "Introduction to Applied Linear Algebra," 2018.

Example: Target localization

- target located at position $(p_\star, q_\star) \in^2$
- agent i knows its position $(p_i, q_i) \in^2$ 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

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

- the target position is the solution to the problem

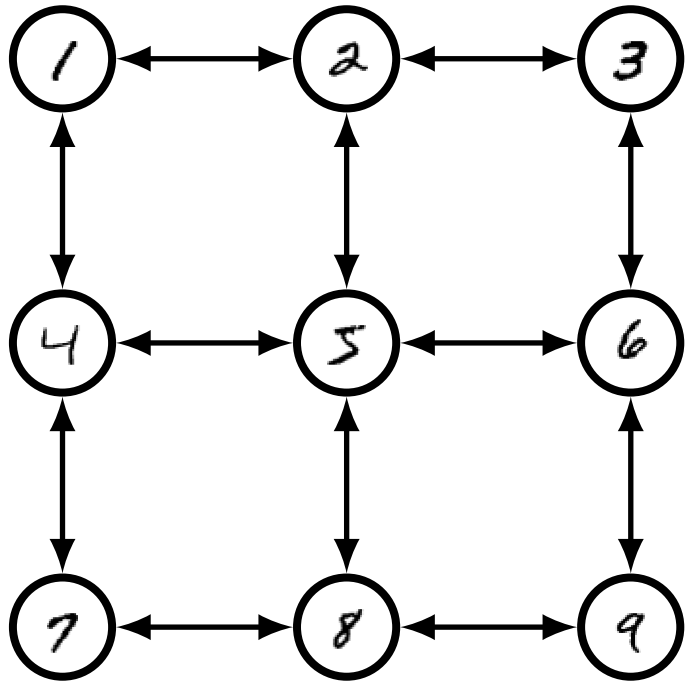
$$\underset{p,\,q}{\text{minimize}} \quad \sum_{i=1}^{n} f_i(p, q)$$

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 $\|y_i - \theta^\mathsf{T}\Phi(x_i)\|$

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

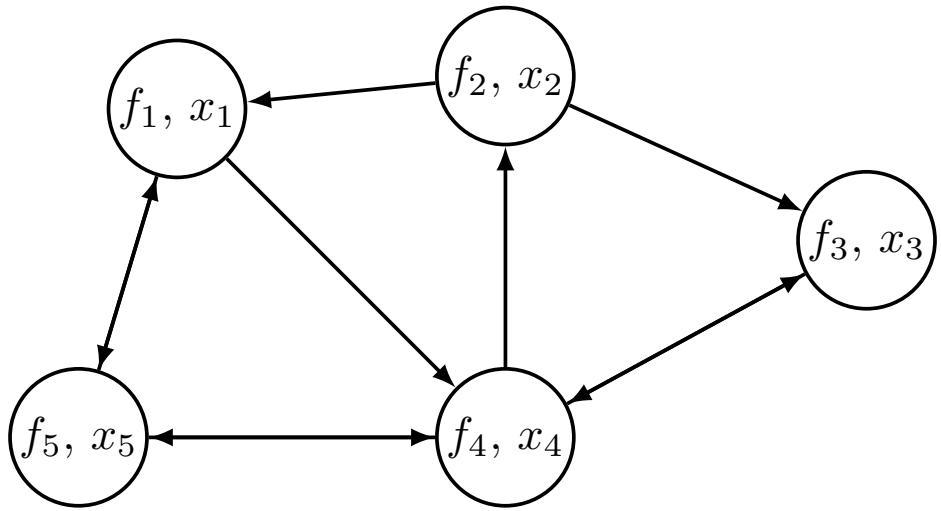


Agents want a *global* model while keeping their local data *private*.

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

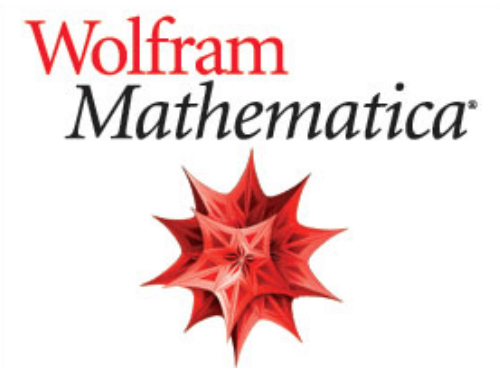
Problem abstraction

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



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

Tools



Contact info

- `https://vanscoy.github.io`
- Garland Hall, Room 260 Q