RBE 510 – Multi-Robot Systems
Lecture 1: Introduction

Kevin Leahy

August 22, 2025

Today

- Introductions
- Syllabus review
- Course overview
- Consensus

Introduction

Kevin Leahy

Assistance Professor, RBE

PhD in Mechanical Engineering, Boston University

kleahy@wpi.edu

Unity Hall 250 D

Research

Formal methods in robotics and autonomous systems

Symbolic planning and control

Safety in AI/ML systems

Course info

- Canvas
  - Assignments, readings, etc. will be posted here
- Time and location:
  - Stratton 311
  - T/F 11:00 AM – 12:50 AM
  - Last day of class October 10
- Grader: Pranay Katyal
- Office hours – TBD
  - Or by appointment

Syllabus

First – A Caveat

- This was revived last A-term after last being taught in 2017
- The course description remains largely accurate
  - No "professional-quality papers"
  - No project
- There will be homework (with programming problems) and an exam

Learning Objectives

- By the end of this course, you students be able to:
  - Model and simulate multi-robot behavior using several different approaches;
  - Analyze the design trade-offs among design choices ranging from centralized, decentralized and distributed architectures in terms of computational complexity and optimality;
  - Implement multi-robot algorithms for agreement, deployment, communication, and decision-making; and
  - Understand the limitations and assumptions underlying many common multi-robot models.

Overview

- Distributed algorithms
  - Agreement
  - Deployment
- Sensing and estimation
- Communication and information sharing
- Cooperative decision making

Assignments and Grading

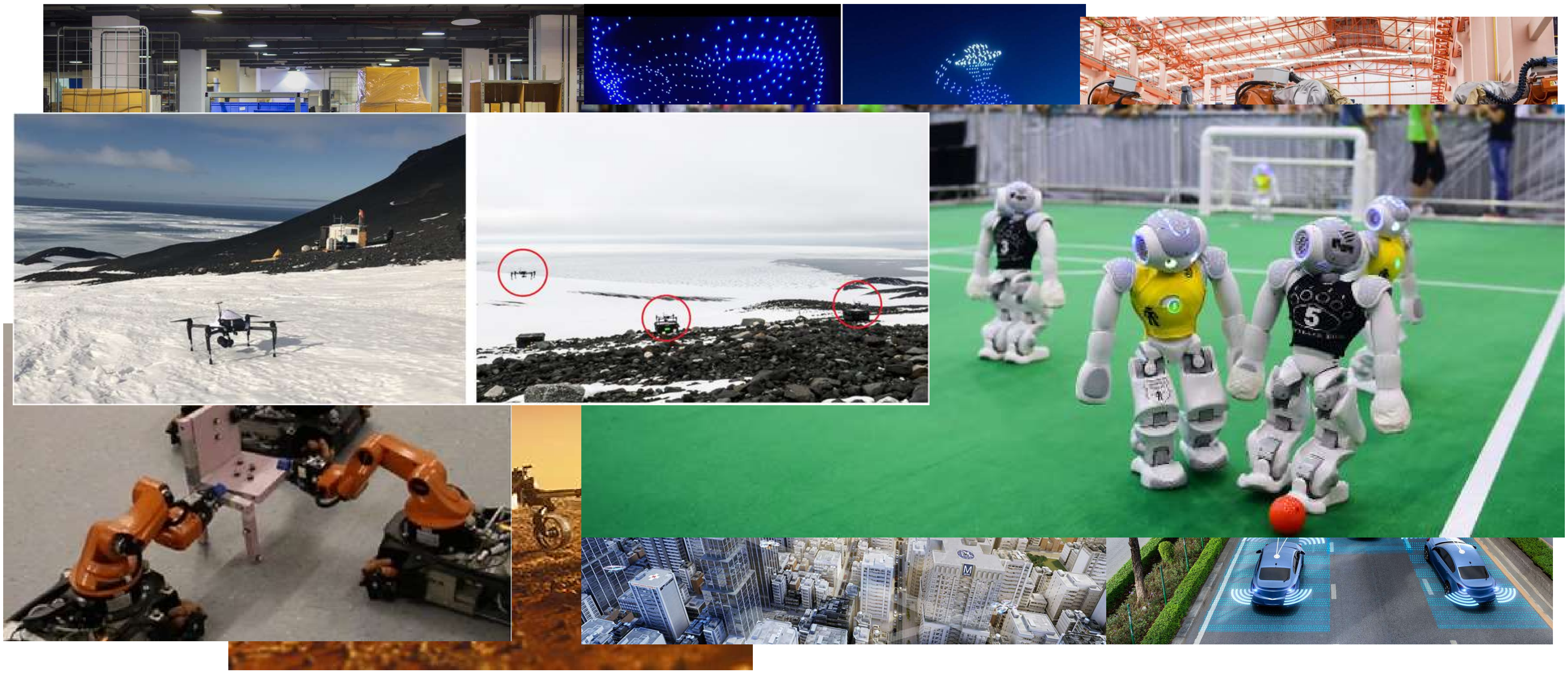
- 5 homeworks, including programming assignments (60%)
  - Plus, homework 0 – late policy applies, but worth 0 points
- Group/pair paper presentations (15%)
- 1 final exam (15%)
- Participation (10%)

Late assignment policy

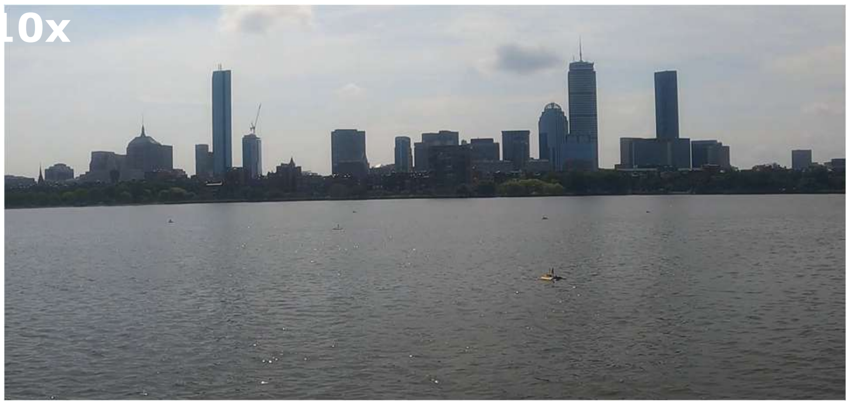
- Late homework assignments will receive a 0
- Lowest homework grade is dropped

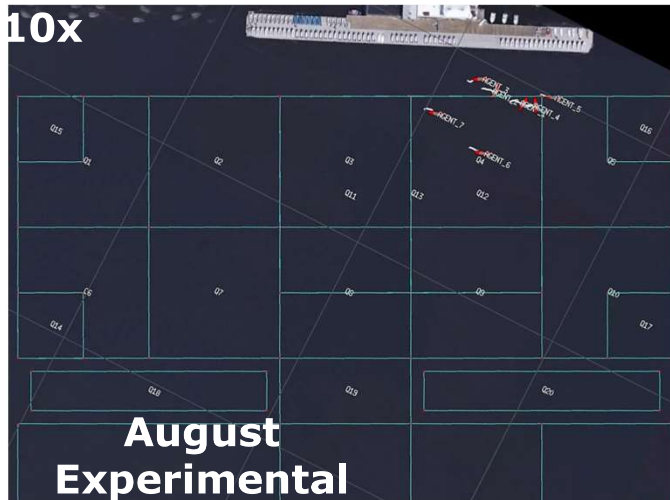
Introduction

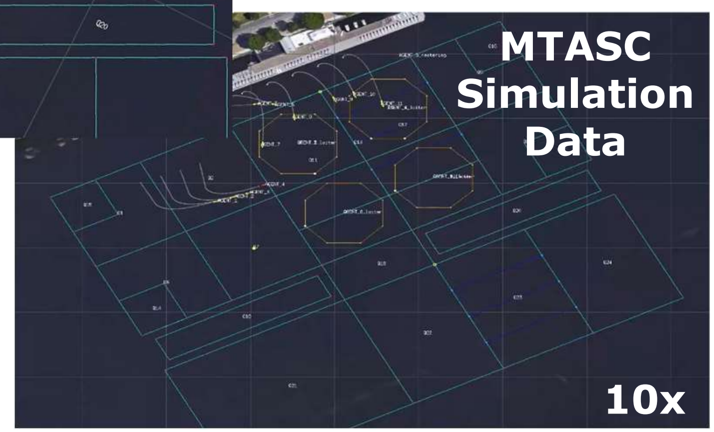
Multi-Robot Systems



Multi-Robot Systems





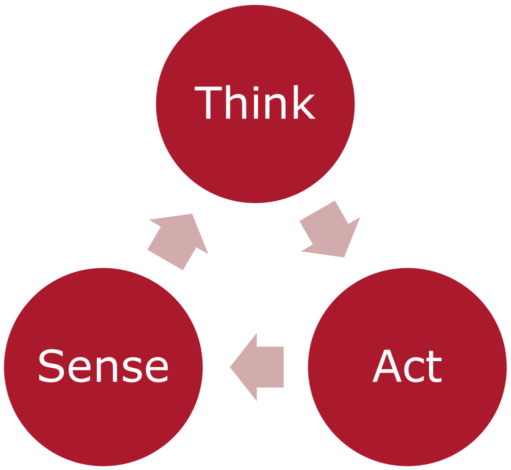


A Very Active Research Area

- 5+ sessions at IROS 2023, and 5+ sessions at ICRA 2024
- Several major events dedicated purely to multi-robot & multi-agent research
- RBE 511 – Swarm Intelligence

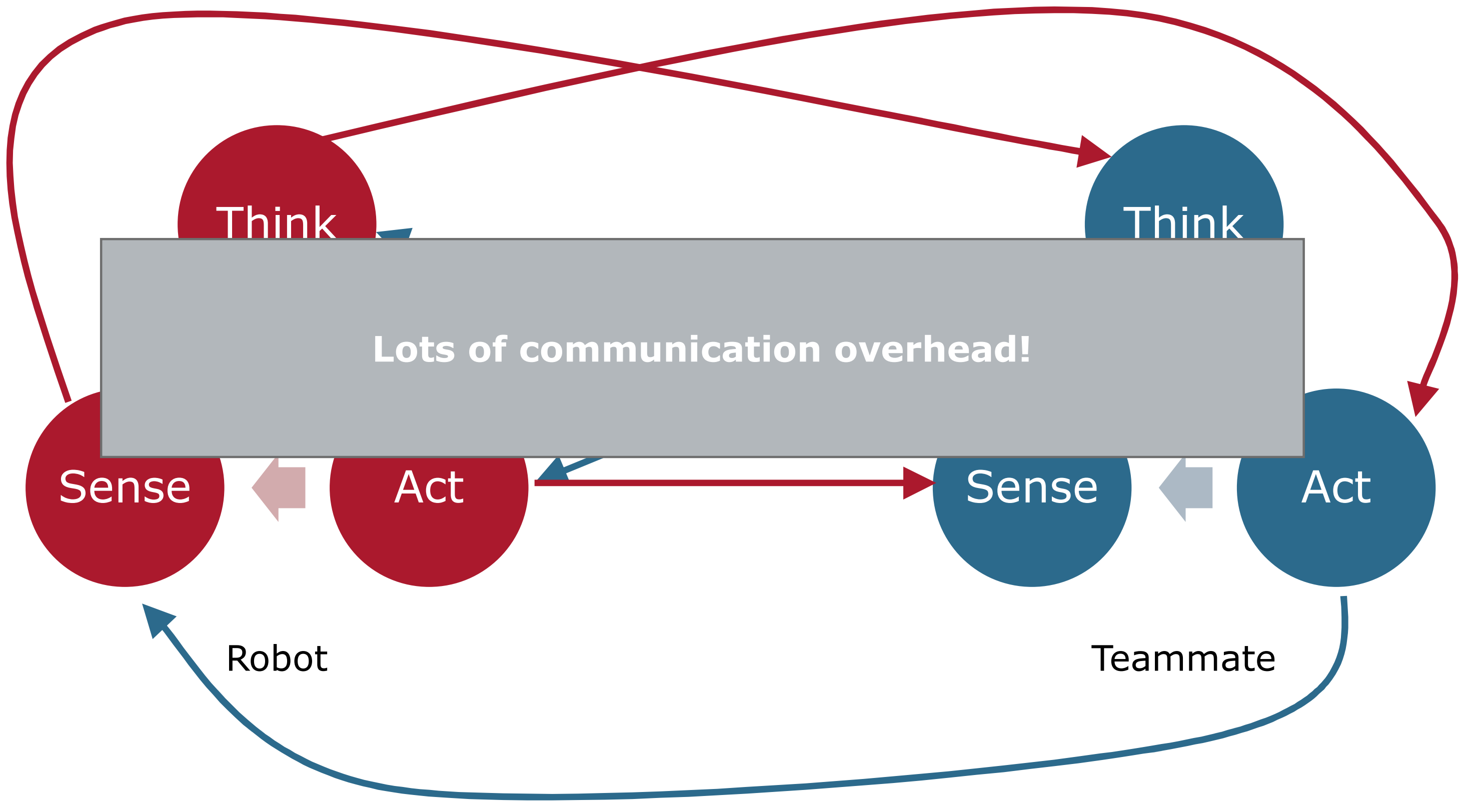
This class can only cover a portion of the space of multi-robot systems

One robot

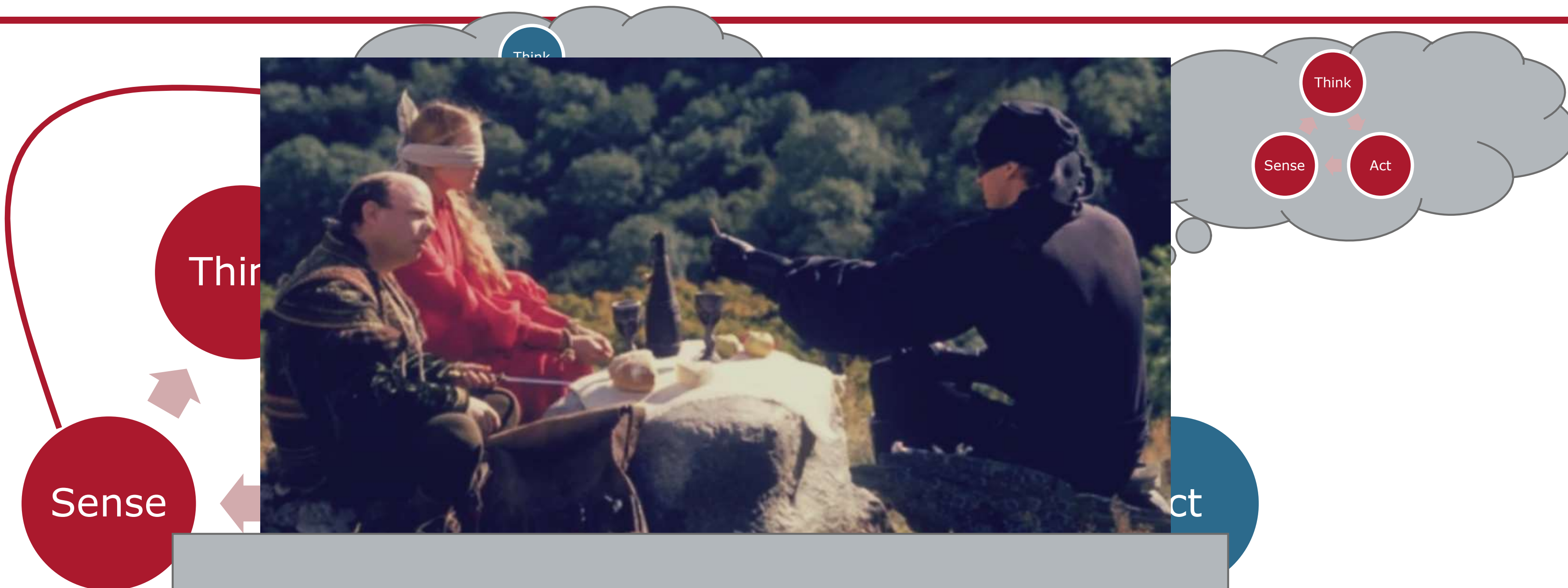


Robot

Multiple robots

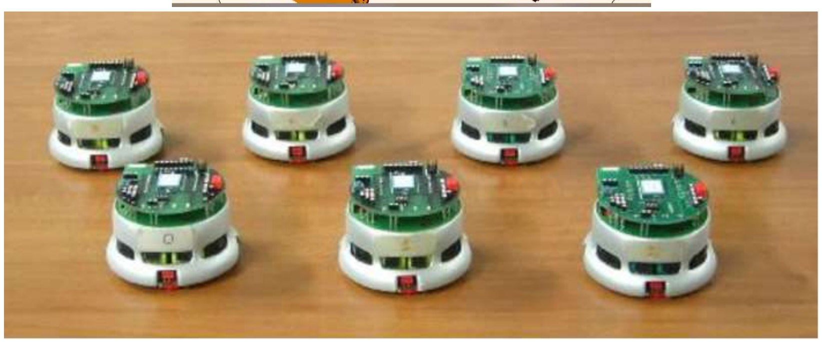


Multiple robots

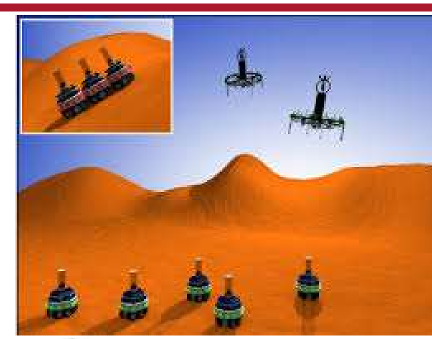


Hard to reason about!

Types of Robots – Physical Differences



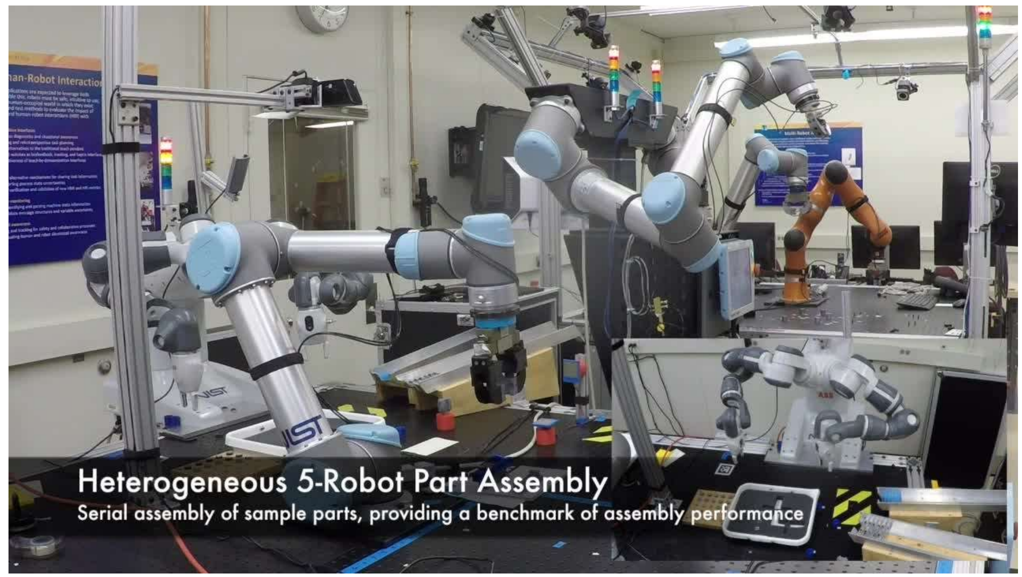
Homogeneous



Heterogeneous

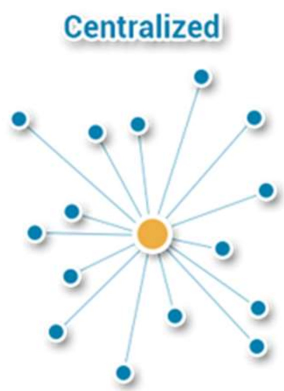
Types of Robots – Interactions

Identical, common task

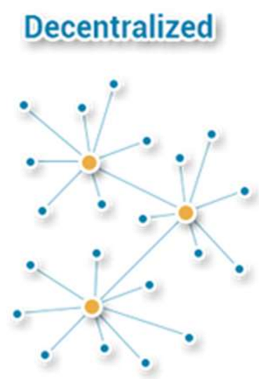


Related, sequential roles

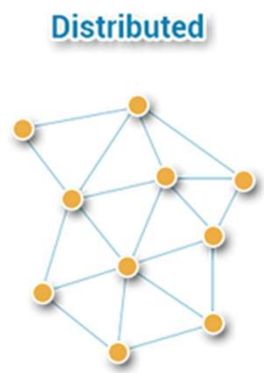
Types of Robots – Architectures



Single authority
Single point of failure
Easy to push out commands
Hard to ingest/process information
Easy to analyze
Hard to Scale



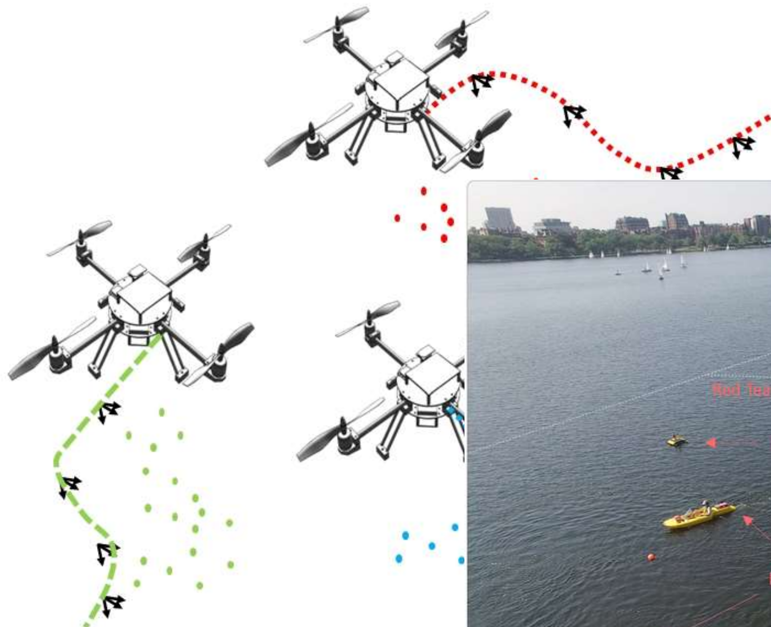
Multiple authorities
Multiple points of failure
Harder to push out commands
Easier to ingest/process information
Harder to analyze
Easier to Scale



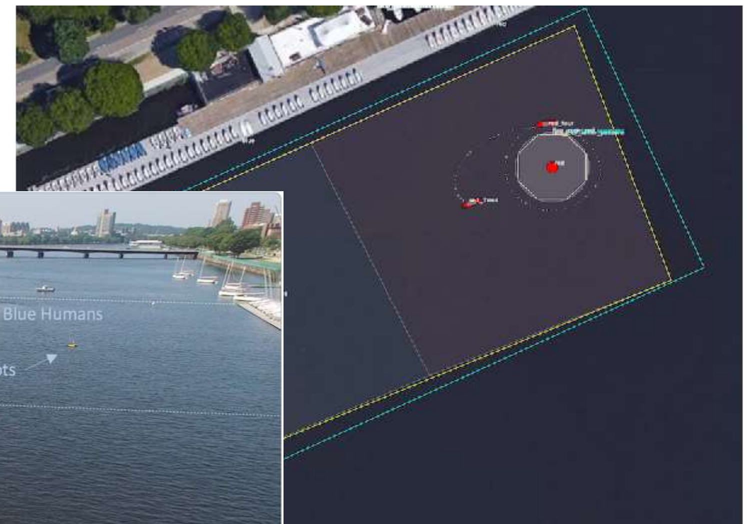
Distributed authority
Robust to failure
Harder to push out commands
Easier to ingest/process information
Hard to analyze
Easy to Scale

What about optimality?

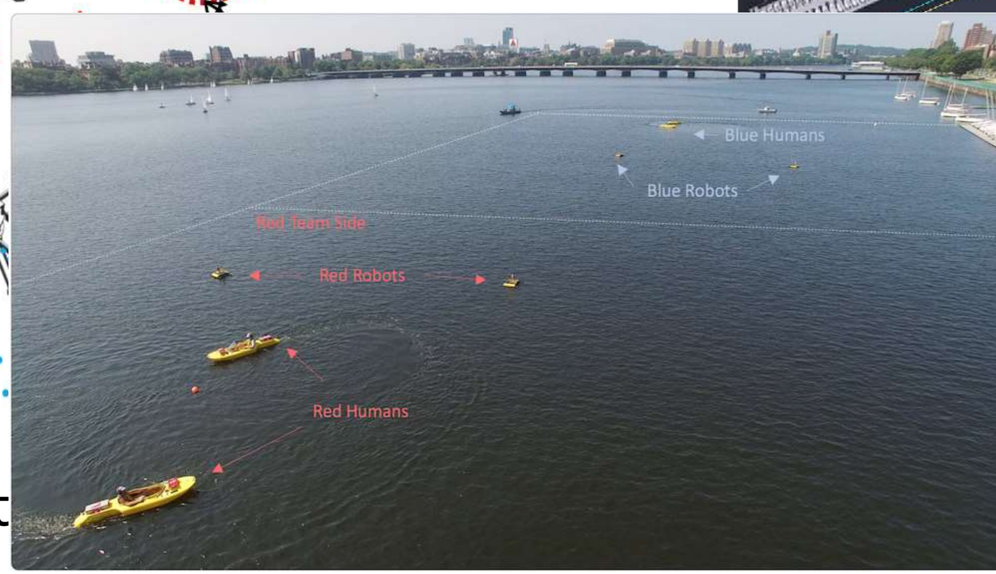
Types of Robots – Behavioral Differences



Cooperat



Adversarial



Mixed

Types of Robots – Behavioral Differences

- Over 2000 citations!

Brian P. Gerkey
Artificial Intelligence Lab
Stanford University
Stanford, CA 94305-9010, USA
gerkey@stanford.edu

Maja J. Matarić
Computer Science Department
University of Southern California
Los Angeles, CA 90089-0781, USA

A Formal Analysis and Taxonomy of Task Allocation in Multi-Robot Systems

Abstract

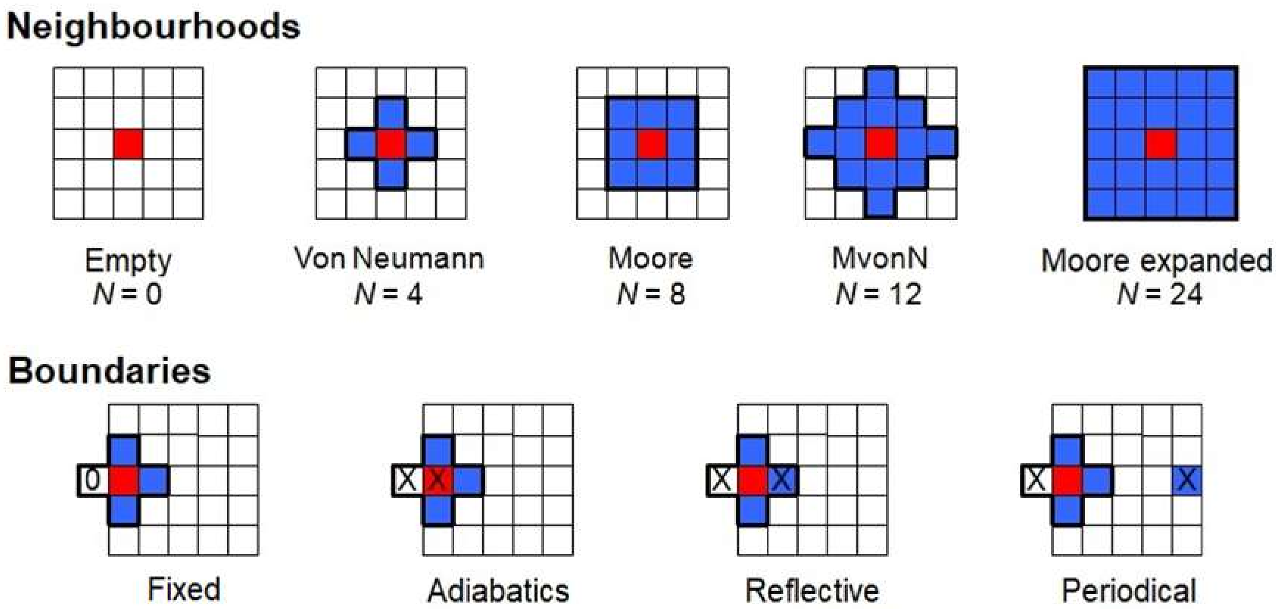
Despite more than a decade of experimental work in multi-robot systems, important theoretical aspects of multi-robot coordination mechanisms have, to date, been largely untreated. To address this issue, we focus on the problem of multi-robot task allocation (MRTA). Most work on MRTA has been ad hoc and empirical, with many coordination architectures having been proposed and validated in a proof-of-concept fashion, but infrequently analyzed. With the goal of bringing objective grounding to this important area of research, we present a formal study of MRTA problems. A domain-independent taxonomy of MRTA problems is given, and it is shown how many such problems can be viewed as instances of other, well-studied, optimization problems. We demonstrate how relevant theory from operations research and combinatorial optimization can be used for analysis and greater understanding of existing approaches to task allocation, and to show how the same theory can be used in the synthesis of new approaches.

significant achievements have been made along these axes, it is no longer sufficient to show, for example, a pair of robots observing targets or a large group of robots flocking as examples of coordinated robot behavior. Today we reasonably expect to see increasingly larger robot teams engaged in concurrent and diverse tasks over extended periods of time.

1.1. Multi-Robot Task Allocation

As a result of the growing focus on multi-robot systems, multi-robot coordination has received significant attention. In particular, multi-robot task allocation (MRTA) has recently risen to prominence and become a key research topic in its own right. As researchers design, build, and use cooperative multi-robot systems, they invariably encounter the fundamental question: which robot should execute which task in order to cooperatively achieve the global goal? By "task", we mean a subgoal that is necessary for achieving the overall goal of the sys-

Interactions Give Rise to Complexity



Jon von Neumann

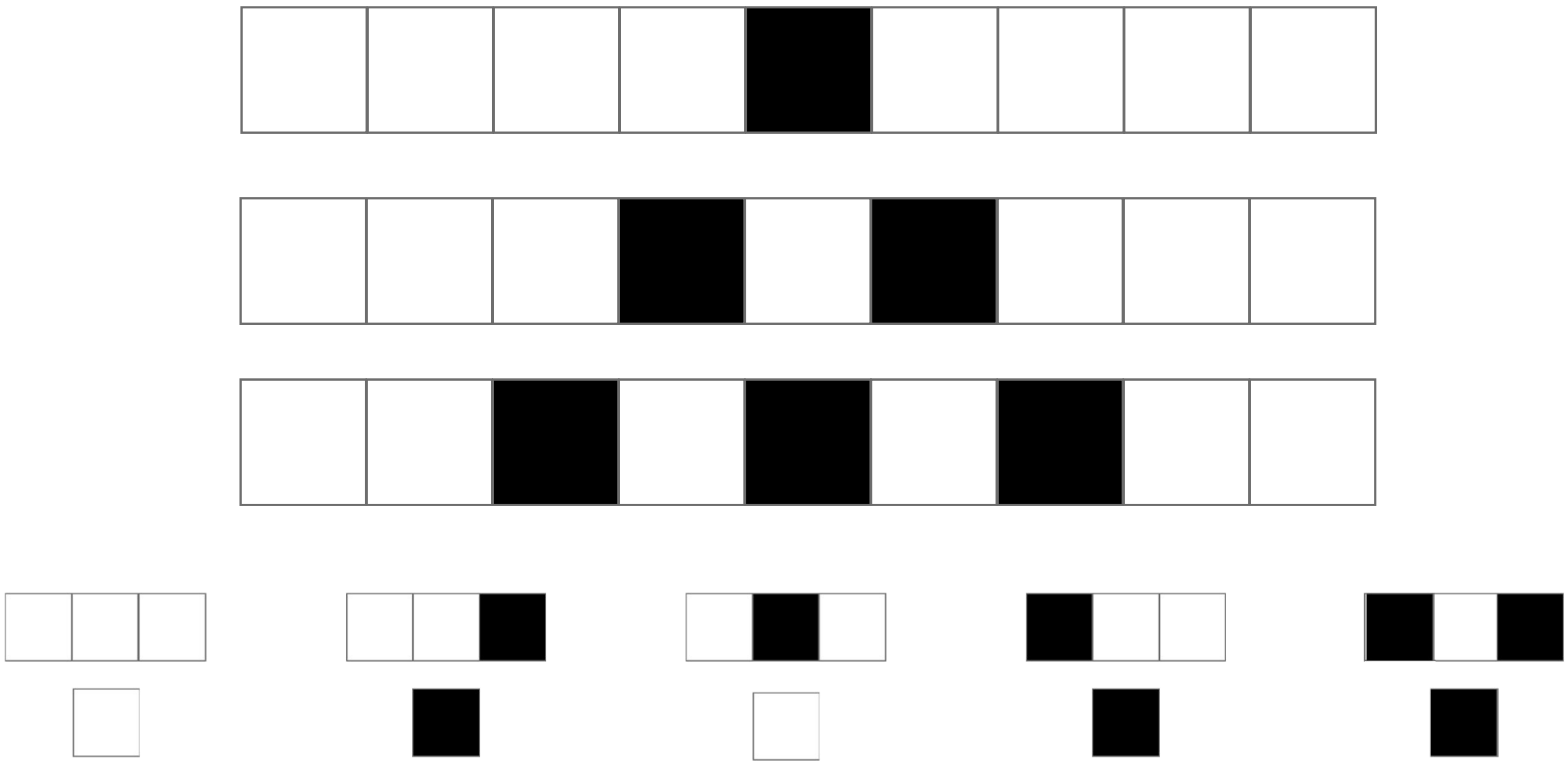
- 1940s at Los Alamos, studying logic of self-replication
- Inspired by crystal lattices and liquid motion

Conway’s Game of Life

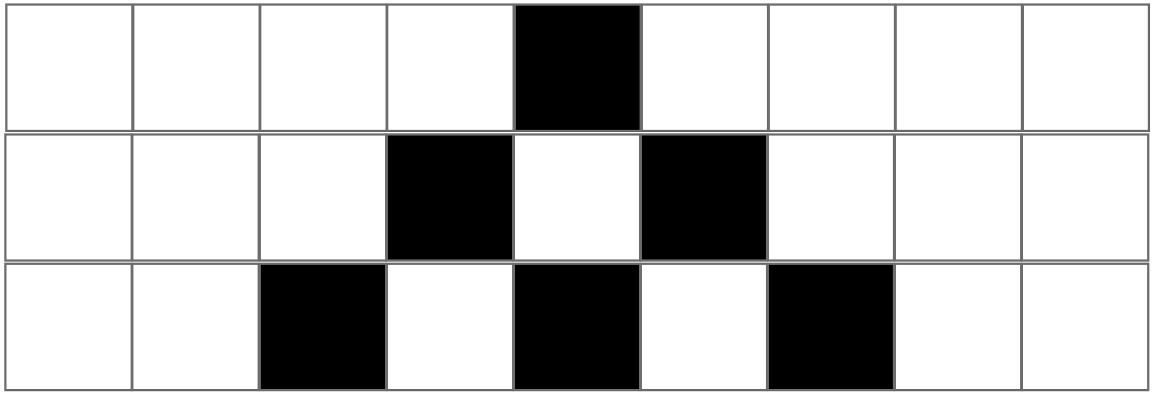
1. Any live cell with fewer than two live neighbours dies, as if caused by underpopulation.
2. Any live cell with two or three live neighbours lives on to the next generation.
3. Any live cell with more than three live neighbours dies, as if by overpopulation.
4. Any dead cell with exactly three live neighbours becomes a live cell, as if by reproduction.

John Conway

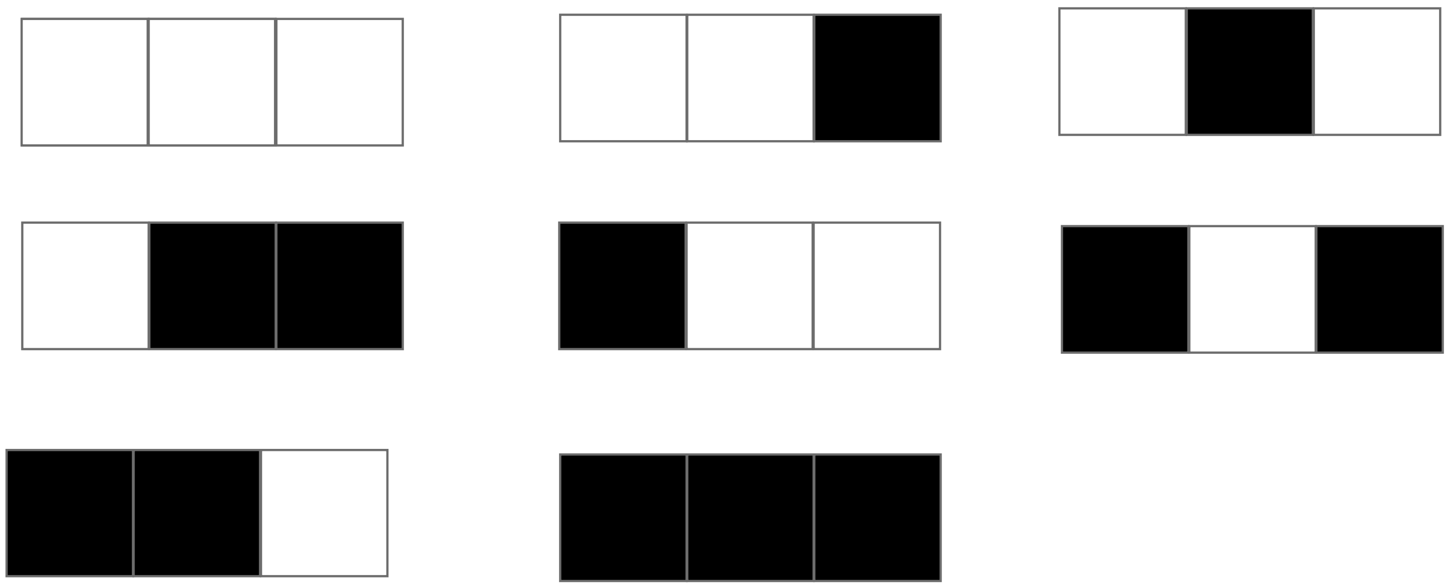
Wolfram’s 1-d Cellular Automata



Wolfram’s 1-d Cellular Automata

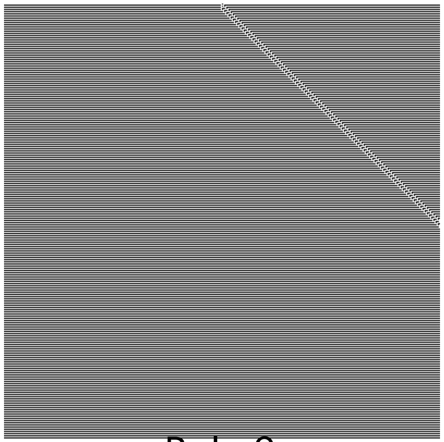


Wolfram’s 1-d Cellular Automata

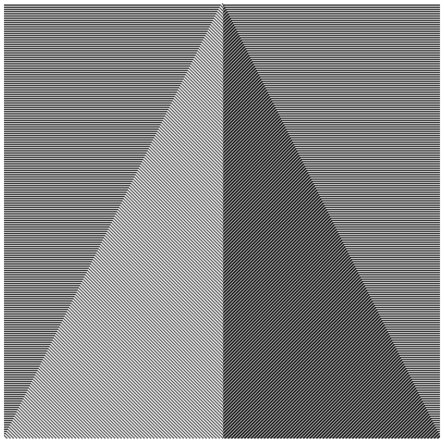


- There are $2^3 = 8$ combinations of states and neighbors
- How many possible automata are there?

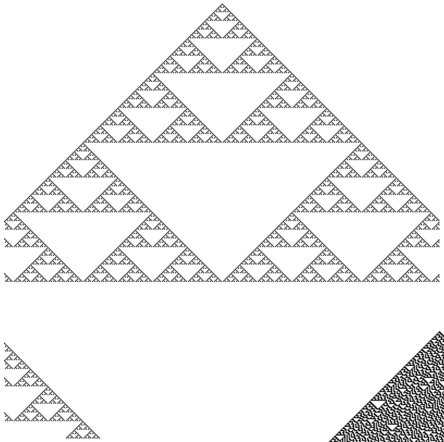
Examples



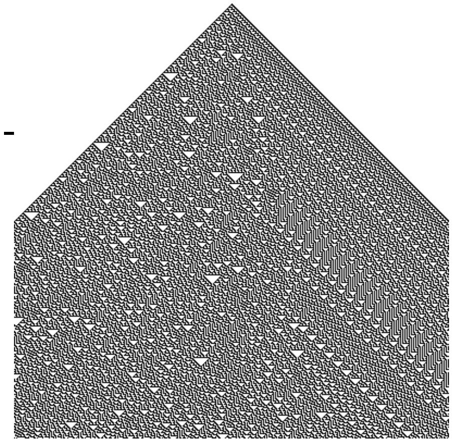
Rule 9



Rule 57

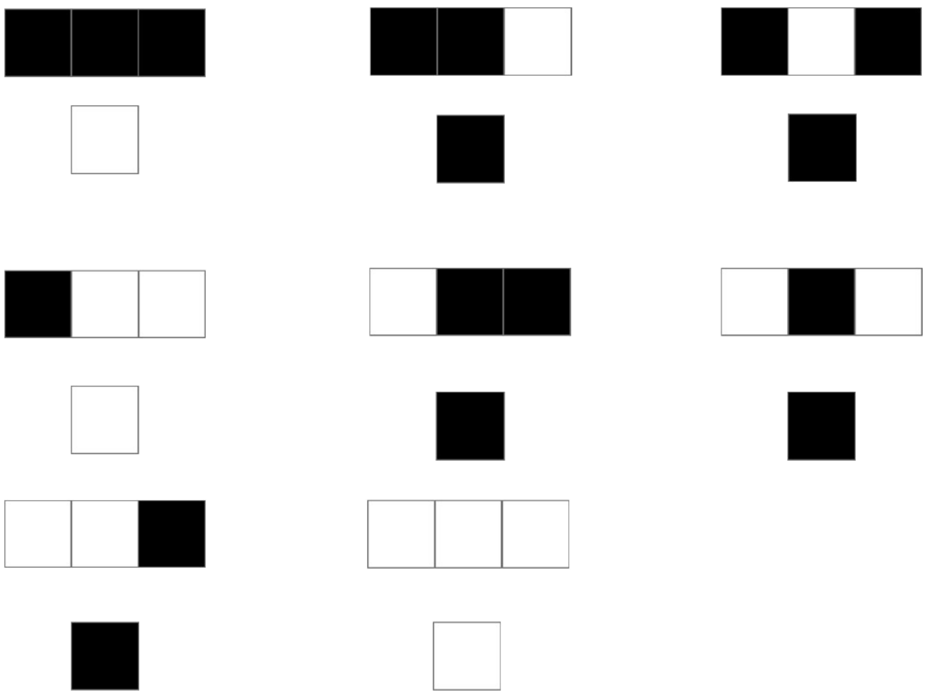


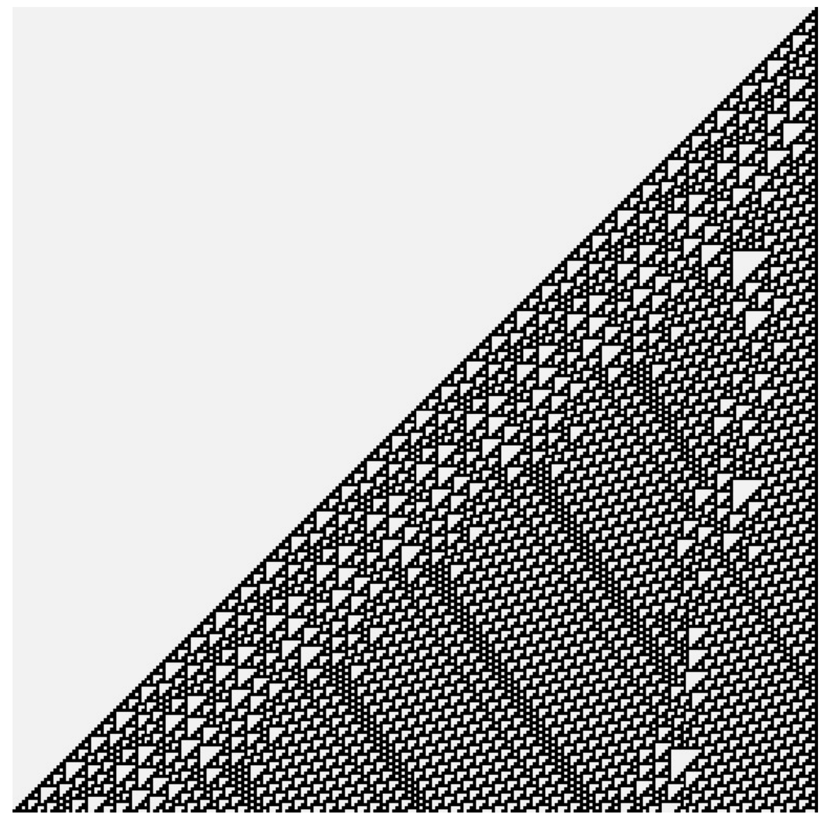
Rule 18 -



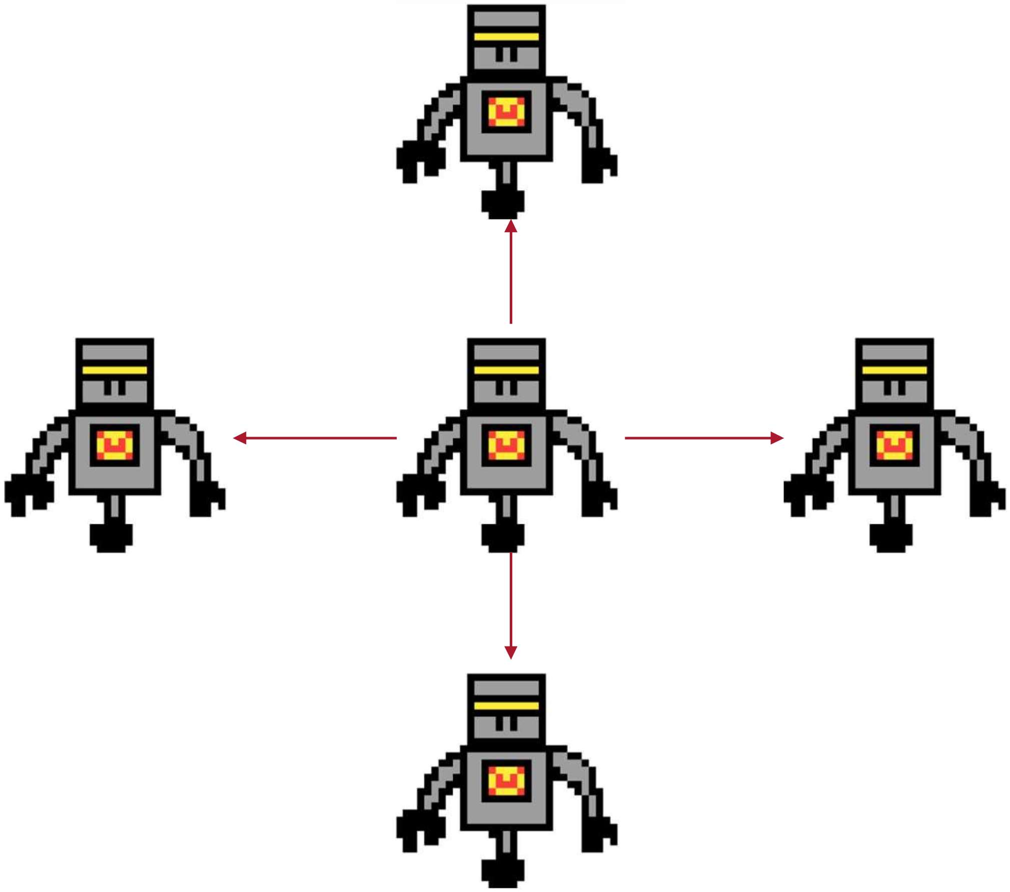
Rule 86

Rule 110



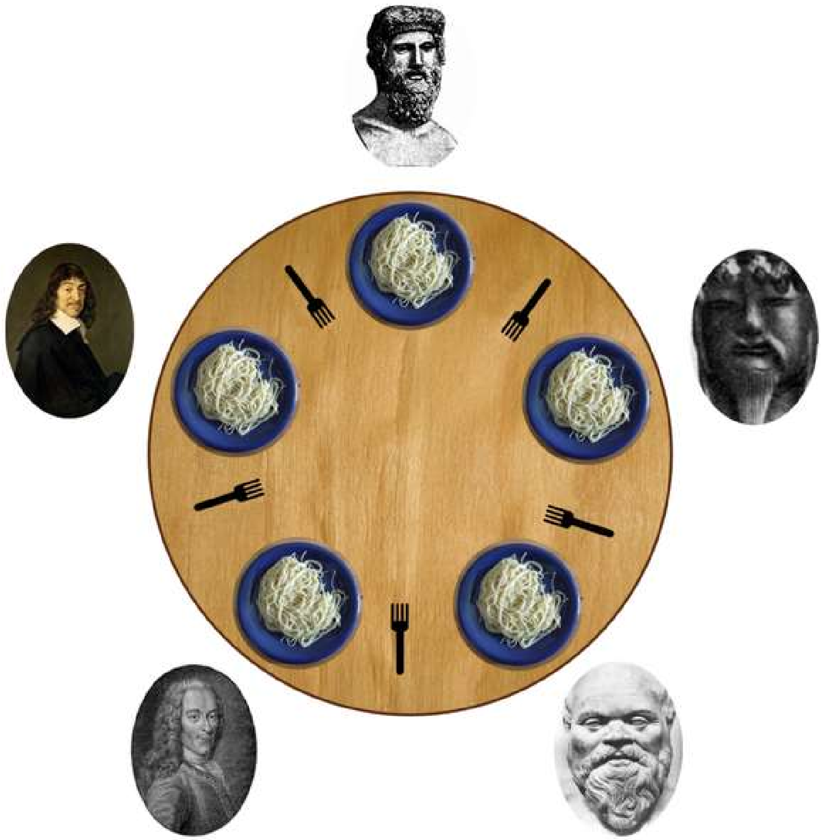


What about robots?



- If this much complexity can arise from binary cells, a lot of complexity can arise from interacting robots!

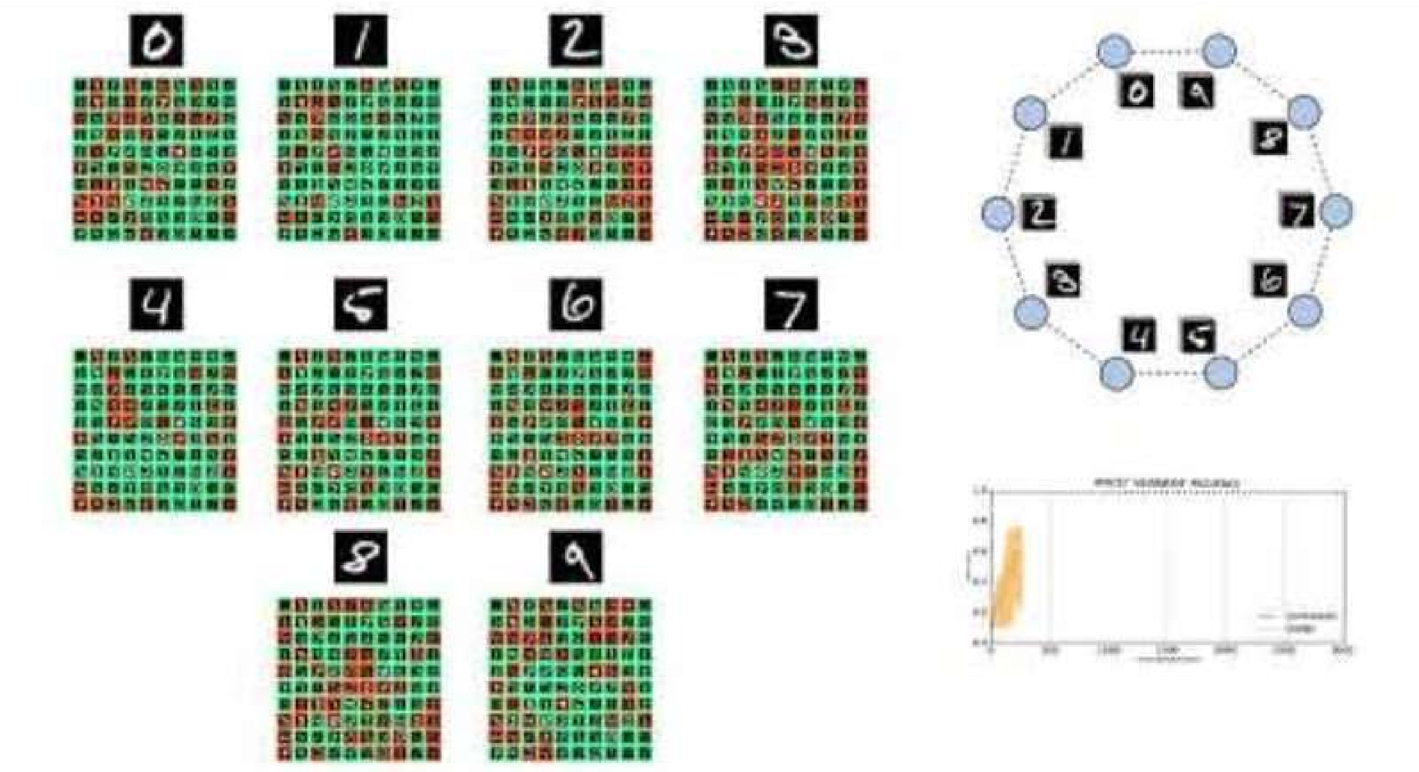
A Very Incomplete History



- 5 philosophers at a table
- 2 forks required to eat
- Each philosopher may either think or eat
- Can only eat when both adjacent forks are available
  - Therefore, when both neighbors are thinking
- After a philosopher eats, they put down both forks
- Goal:
  - Design algorithm so that each philosopher can eat and think repeatedly, without coordinating with others

A Very Incomplete history

A Very Incomplete history



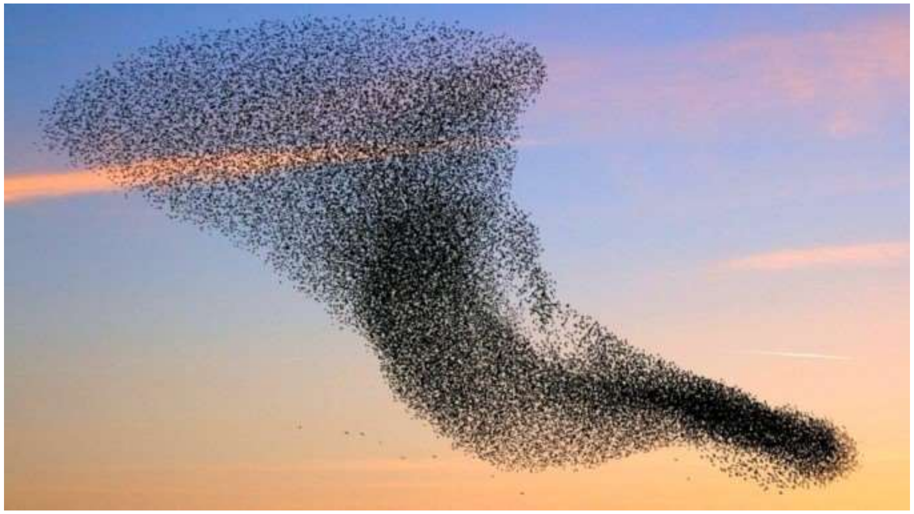
Guiding Questions

- Who needs to talk to whom?
  - How often?
  - How much information?
  - What information?
- Who knows what?
  - How do they know it? Comms? Sensing?
  - Local information? Global information?
- Who decides what?
  - Based on what information?
  - How are decisions distributed?

Mathematical Background and Consensus

Example Behaviors

Flocking

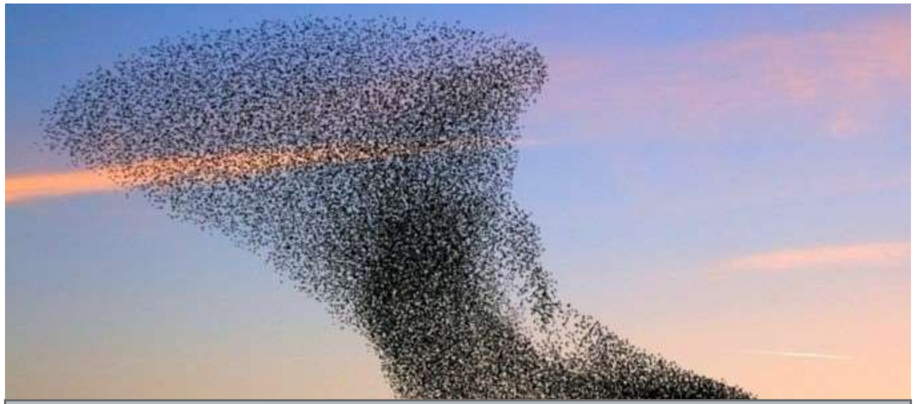


Rendezvous

What do these two behaviors have in common?

Example Behaviors

Flocking



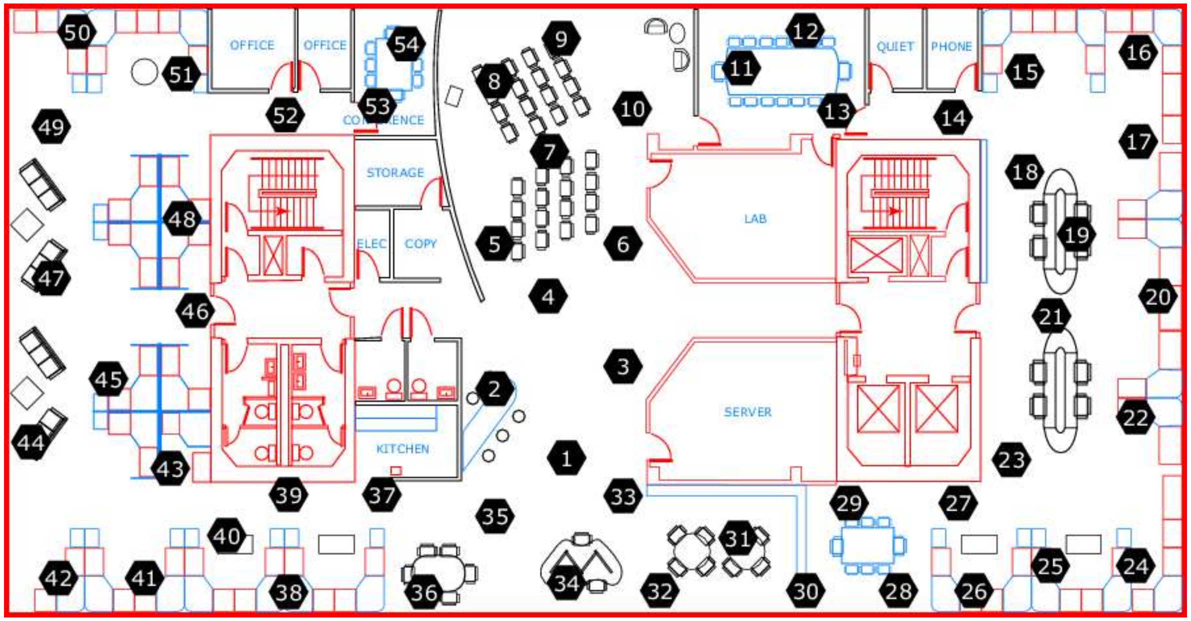
Agree on heading with neighbors

Rendezvous

Agree on location with neighbors

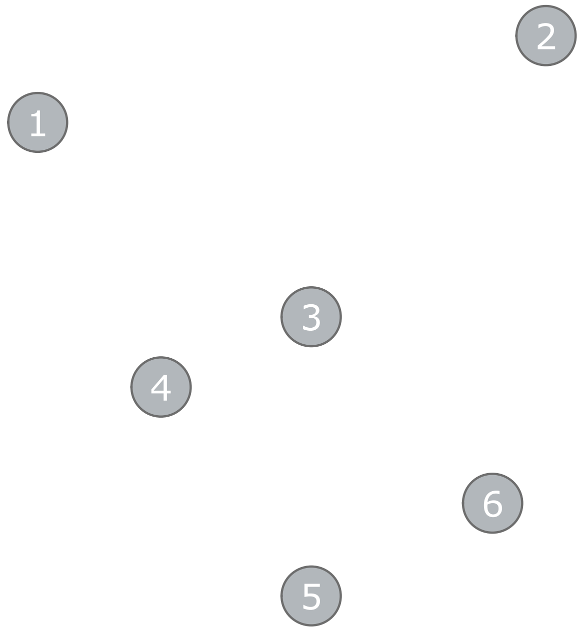
Consensus!

An example – temperature sensing



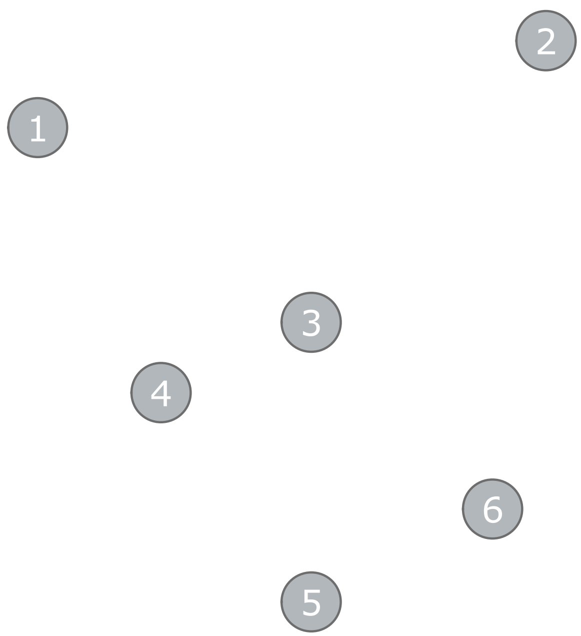
How would we determine the true temperature in the room?

Computing averages



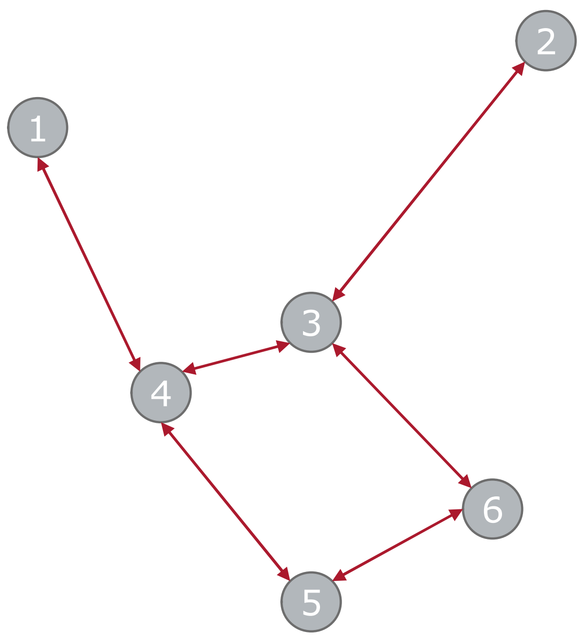
- We have 6 sensors
- Each measures temperature in its own neighborhood
- How do we compute the average temperature?

Linear Consensus Protocol



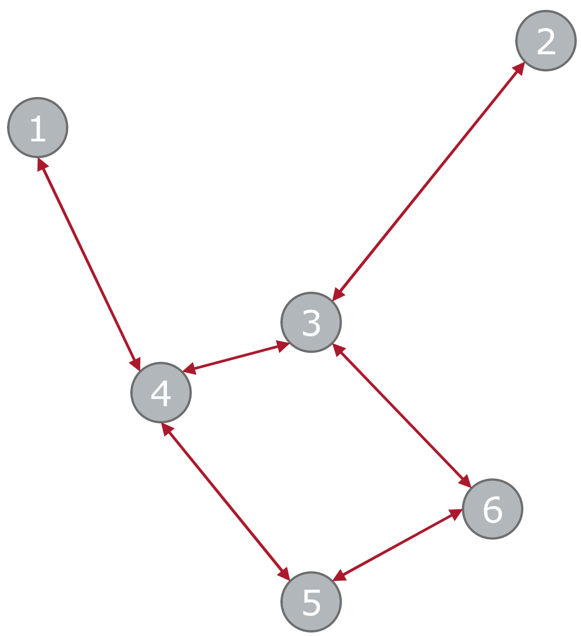
- We have I agents
  - Here, $I = \{1,2,3,4,5,6\}$
- Generic agent is agent i
  - Agent i has state x_i
  - Initial state of agent i is x_i^0
- Want all agents to agree on common state
  - e.g., states converge to $\bar{x} = \frac{1}{6}\sum_1^6 x_i$
- How?

Linear Consensus Protocol



- Agents need to share information
- Agents communicate if they are "close enough"
  - We'll talk more about this in future lectures
- Represent this as a graph
  - Edge between two nodes says that they can exchange information

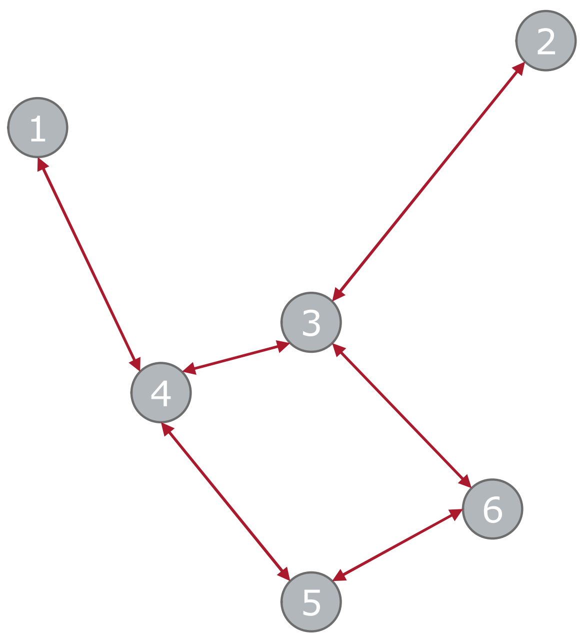
Graphs



- Graph $G = (V, E)$
- V is a set of *nodes*
- $E \subseteq V \times V$ is a set of *edges*

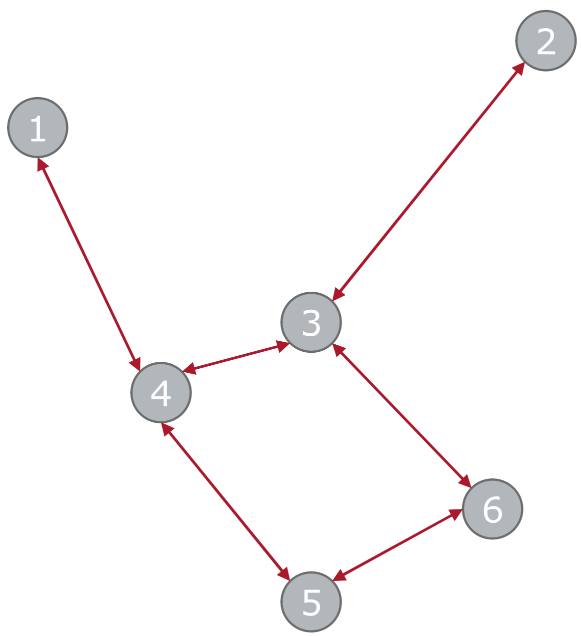
- This graph is *undirected*
  - If agent i can communicate with agent j, then agent j can communicate with agent i
  - This is not true of all graphs!

Graphs



- Graph $G = (V, E)$
- V is a set of *nodes*
- $E \subseteq V \times V$ is a set of *edges*
- What is V for this graph?
- What is E?

Graphs

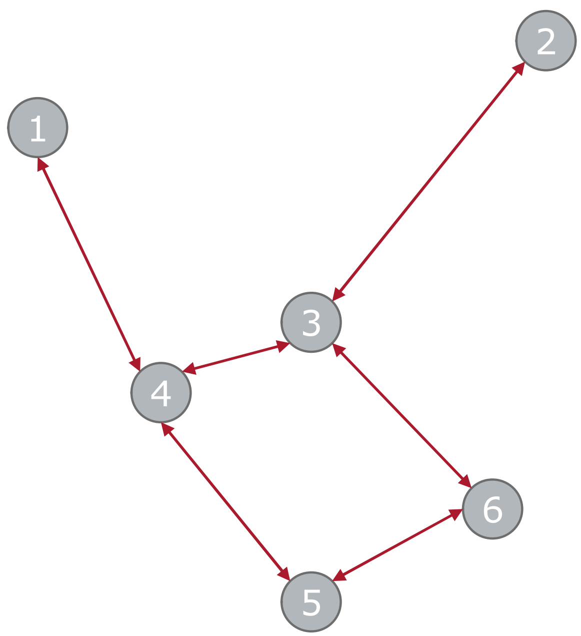


- Neighborhood (or neighbor set)
  - $\mathcal{N}_i = \{ j \mid (i,j) \in E \}$
  - Degree of i is $|\mathcal{N}_i|$

- What is the degree of node 1?

- What is the degree of node 3?

Back to Consensus



- How do we get agents to agree on a value?
- In control problems, we use an *error* signal
  $$e = x_d - x$$
- But we don't know x_d
- How about our *disagreement** instead?
  $$x_i - x_j$$

*Note: there is also a *disagreement* function, which we're not really concerned with here

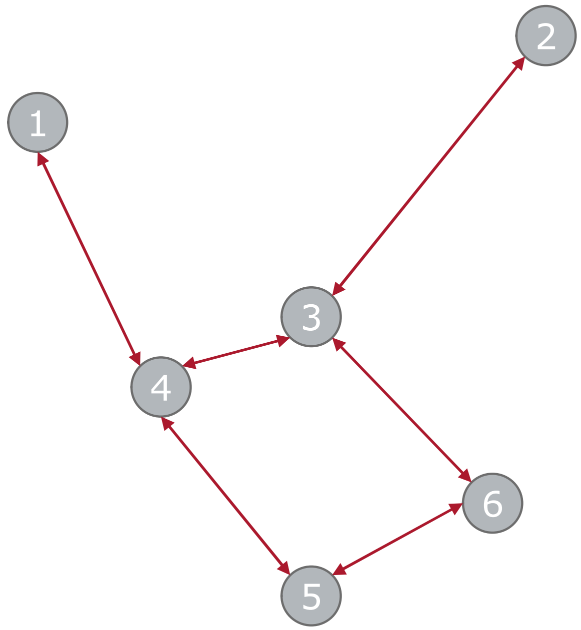
Linear Consensus Protocol

$$\dot{x}_i(t) = \sum_{j \in \mathcal{N}_i} \left(x_{j(t)} - x_{i(t)}\right)$$

$$x_i(t+1) = x(t) + \alpha \sum_{j \in \mathcal{N}_i} \left(x_j(t) - x_i(t)\right)$$

- A nice feature – agents only need information from their neighbors (i.e., *local* information)
- But we care about *global* behavior
- How can we link the two?

Algebraic Graph Theory

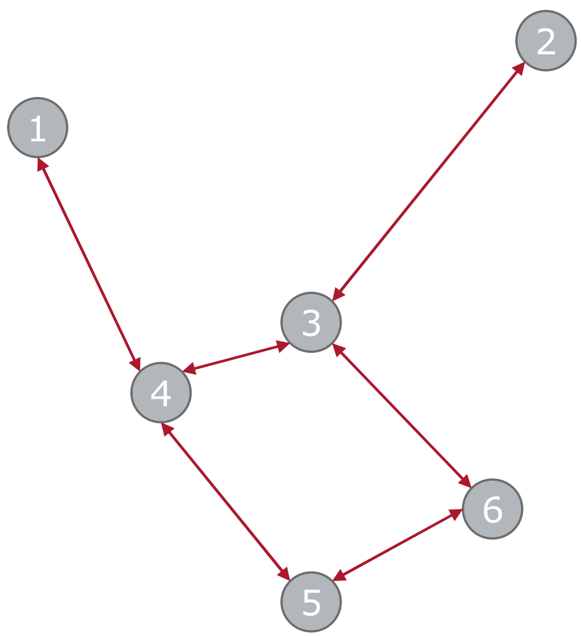


- We can use matrices to make our lives easier!

- Degree matrix

$$D = \begin{bmatrix} |\mathcal{N}_1| & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & |\mathcal{N}_n| \end{bmatrix}$$

Algebraic Graph Theory



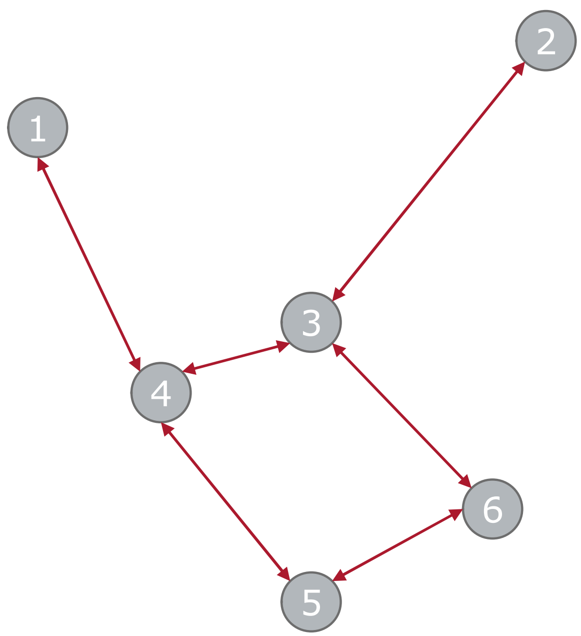
- Adjacency relationship

$$a_{ij} = \begin{cases} 1 & if\ (i,j) \in E \\ 0 & otherwise \end{cases}$$

- Adjacency matrix

$$A = \begin{bmatrix} 0 & \cdots & a_{ij} \\ \vdots & \ddots & \vdots \\ a_{ji} & \cdots & 0 \end{bmatrix}$$

Graph Laplacian



- *Laplacian matrix* of a graph is
$$L = D - A$$

- What is L for this graph?

Laplacian Matrix – Fun Facts

- Laplacian for undirected graph is real-valued and symmetric (why?)

- Eigenvalues are real, non-negative
- Eigenvectors are real, orthogonal

- It holds that at least one eigenvalue is 0, i.e.,

$$0 = \lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_n$$

Laplacian Matrix – Fun Facts

- The sum of a row is always zero (why?)
- This means that $L\mathbf{1} = 0$, where $\mathbf{1} = [1, 1, \ldots, 1]^T$
- $\mathbf{1}$ is an eigenvector
- 0 is an eigenvalue

Consensus and the Laplacian

$$\dot{x}_i(t) = \sum_{j \in \mathcal{N}_i} \left(x_{j(t)} - x_{i(t)}\right)$$

- Let's stack all the agents together s.t. $\boldsymbol{x} = [x_1, x_2, \ldots, x_n]^T$

- Then,

$$\dot{\boldsymbol{x}}(t) = -L\boldsymbol{x}(t)$$

3 Agent Example – Laplacian

Linear Consensus Protocol – Intuition

$$\dot{x}_i(t) = \sum_{j \in \mathcal{N}_i} \left(x_{j(t)} - x_{i(t)} \right)$$

$x_i - x_j$: how far away x_i is from $x_i = x_j$

Negating $x_i - x_j$ moves the system “in the direction of j”

Consensus to the Average and some Proofs

We are cheating here!

- But only a little bit
- We (as designers) use our *global* view of the system
- Agents only have *local* information

Wrap-Up

Recap

- Syllabus
- Overview
- Linear consensus protocol

Next Time

- Modeling multi-robot systems
  - Motion
  - Comms
  - Computation

- Agreement Protocol Continued
  - Formation Control
  - Time-varying topologies