Lecture 1: Introduction

Teacher:
Gianni A. Di Caro
Group of ants carrying a big prey (lizard) to colony nest
Ants fill a gap by growing from both sides an ant bridge
Bird flocking, smooth, mesmerizing
Boids flocking! (from C. Reynolds’ work, 1986)
Fish schooling (sardines)
Collective Intelligence?

Fireflies synchronized flashing (in Thailand)
Two-ways and three-ways bridge (homemade) settings for ant colonies: pheromone trails allow the colony finding the shortest way
Ant colonies are able to discover short-cuts!
Swarms of locusts: flocking, leader follower behaviors
Swarm of robots dynamically self-organizing
Pedestrians and mobile robots creating ordered flows
COLLECTIVE INTELLIGENCE?

Complex systems

- Multi-component
- Multi-agent
- Usually $\gg 1$
- Emergence
- Self-organization
- Phase transitions
- Multiple equilibria / solutions
- Distributed
- Decentralized
- Dynamic
- Time evolution
- Evolutionary pressure

Modeling? Properties? Control?
COMPLEX ≠ COMPLICATED
Emergence:
A system exhibits emergence when there are coherent emergents at the macro-level that dynamically arise from the (localized) interactions between the parts at the micro-level. Such emergents are novel w.r.t. the individual parts of the system.

“Precursors” of the notion of emergence:
- Whole before its parts
- Gestalt: a configuration or pattern of elements so unified as a whole that it cannot be described merely as a sum of its parts
- Non-linearity

\[ f(a+b) \neq f(a) + f(b) \]
**EMERGENCE: SPACE-TIME SCALES**

- What are the **micro** and **macro levels**?
- The **Observer** matters: space-time scales
**Radical novelty:** The collective behavior is not readily understood from the behavior of the parts. The collective behavior is, however, *implicitly contained* in the behavior of the parts if they are studied in the context in which they are found.

Emergent properties cannot be studied by physically taking a system apart and looking at the parts (=*reductionism*). They can, however, be studied by looking at each of the parts in the context of the system as a whole.

**Problem:** Predictability of the behavior / evolution of the system!?
Emergence: Decentralized Control

- Only *local mechanisms* to influence the global behavior.
- There is no central control, i.e. no single part of the system directs the macro-level behavior: The actions of the parts are controllable. The whole is not directly controllable → No leader!
- This characteristic is a direct consequence of the radical novelty that is required for emergence. Centralized control is only possible if that central part of the system has a representation of the global behavior (e.g. a plan)
- Not even centralized instruction sets with a representation of the global behavior (e.g., set provided by an ‘orchestra’ director)
**Self-Organization**

Self-organization is a dynamical and adaptive process where systems acquire and maintain structure themselves, without external control.

The structure can be spatial, temporal or functional.
Complex Systems: Fingerprints

- Multi-agent / Multi-component
- Decentralized: neither central controller, nor representation of global patterns/goals
- Possibly (not necessarily) with a large number of components
- Localized interactions (allowing propagation of information)
- Emerging and / or Self-Organizing properties
- Agents do not need to be “complex”
- Dynamic: Time and space evolution of the system
A view of Complex Systems

Characteristics of Complex Systems

- A 'complex' system
- Emergent behavior that cannot be simply inferred from the behavior of the components

Complex Systems
- Involve:
  - Many Components
  - Dynamically Interacting and giving rise to
  - A Number of Levels or Scales which exhibit
  - Common Behaviors

A 'simple' system

Hierarchies

Self-Organization

Control Structures

Composites

Substructure Decomposability

Emergence

Trandisciplinary Concepts

Across Types of Systems, Across Scales, and thus Across Disciplines
Goals:

- Understanding and modeling natural systems for prediction and control
- Design artificial systems that enjoy complex systems properties

Properties (that could be obtained):

- Robustness
- Parallelism
- Adaptivity
- Fault-tolerance

A bottom-up way of “computing” ....
**Bottom-up vs. Top-Down**

- **Bottom-up programming:** instantiate the single (possibly relatively simple) components, their interaction protocols and topology, local information exchange, (hope) self-organization ... → Output: (Useful) Emerging patterns and functionalities (if any), likely not “precise” but robust, scalable, adaptive

- **Top-down programming** (”regular programming”): write precise instructions for the task to be executed in a variant of classical VonNeumann architecture
BOTTOM-UP CHALLENGES

Given a task/problem to deal with, a number of design choices:

1. Characteristics/skills of the agents
2. Size of the population (related to the choice 1.)
3. Neighborhood definition
4. Interaction protocols and information to exchange
5. Where the information is updated (agent, channel, environment)
6. Use or not of randomness (or, heuristic decisions)
7. Synchronous or asynchronous activities and interactions
8. ...

Lots of parameters

Predictability and efficiency are important issues

Is a top-down approach better?
## OUR SET OF COLLECTIVE INTELLIGENCE TOPICS

<table>
<thead>
<tr>
<th>Week</th>
<th>Topics</th>
</tr>
</thead>
</table>
| 1    | **Introduction:** Complexity, networks, self-organization, emergence, multi-agents, swarms  
**Dynamical systems:** Population models, differential equations  
**Dynamical systems:** Stability, bifurcations |
| 2    | **Dynamical systems:** Discrete-time recurrence equations, logistic model  
**Dynamical systems:** Deterministic chaos |
| 3    | **Cellular automata:** Discrete-time spatial models, properties  
**Cellular automata:** Pattern formation, applications |
| 4    | **Networks:** Topological properties, topology and network computation  
**Networks:** Information diffusion, gossip protocols, topology and CA |
| 5    | **Swarm intelligence:** Dynamic topology, Particle Swarm Optimization (PSO) for optimization  
**Swarm intelligence:** PSO for navigation and foraging in robot swarms |
| 6    | **Swarm intelligence:** Stigmergic communications, Ant-inspired algorithms  
**Swarm intelligence:** Ant Colony Optimization (ACO) |
| 7    | **Task allocation:** Division of labor in animal societies, ant-inspired clustering  
**Task allocation:** Market-based models |
<table>
<thead>
<tr>
<th></th>
<th>Topic</th>
</tr>
</thead>
</table>
| 8 | **Social decision-making and consensus:** Crowd wisdom, quorum responses, leaders/followers  
**Social decision-making and consensus:** Distributed consensus |
| 9 | *Spring break*                                                        |
| 10| **Social decision-making and consensus:** Social choice, theory of voting,  
**Social decision-making and consensus:** Designing voting rules |
| 11| **Game theory:** Minimax theorem, Equilibria concepts  
**Game theory:** Correlated equilibria, Leader-follower games |
| 12| **Game theory:** Evolutionary games  
**Game theory:** Social welfare, Optimality models |
| 13| **Pattern formation:** Moving in a crowd  
**Pattern formation:** Flocking |
| 14| **Neural models:** Neural networks, perceptrons  
**Neural models:** Self-organizing maps |
| 15| **Neural models:** Self-organizing maps in deep learning  
Case study, Q&A |
DYNAMICAL SYSTEMS

\[ \frac{dx}{dt} = \dot{x} = X(x) \]
\[ x_{t+1} = f(x_t) \]

\[ \dot{N}(t) = \frac{dN}{dt} = rN(t) \left(1 - \frac{N(t-T)}{K}\right) \]

Time evolution (depending on initial conditions)

Attractors

Bifurcations, dependence on parameters
DYNAMICAL SYSTEMS

Basic ingredients:

State variables: $\mathbf{x} = (x_1, x_2, \ldots, x_n)$

Evolution operator

$$x_i(t) = \Phi_i(t; x_1(t_0), \ldots, x_n(t_0)) \quad i = 1, \ldots, n \quad t \geq 0$$

Initial condition $x_i(t_0) \ i = 1, \ldots, n$ + Local evolution laws

In continuous time: $t \in \mathbb{IR}$, differential equations

$$\frac{dx_i(t)}{dt} = f_i(x_1(t), x_2(t), \ldots, x_n(t)) \quad i = 1, \ldots, n$$

In discrete time: $t \in \mathbb{IN}$, difference equations (iterated maps)

$$x_i(t + 1) = f_i(x_1(t), x_2(t), \ldots, x_n(t)) \quad i = 1, \ldots, n$$

These are systems of first order autonomous evolution equations.

We shall see how higher order equations, as well as nonautonomous equations can be reduced to this kind of evolution equations.
CELLULAR AUTOMATA

Time-discrete spatially-dependent dynamical systems

Capable of universal computation
CELLULAR AUTOMATA
Connectivity matters: Information dissemination, gossiping, epidemic models

Random, small-world, scale-free topologies

Centrality measures

PageRank
**Social neighborhood**

**Stigmergy**

Particle Swarm Optimization (PSO)

Ant Colony Optimization (ACO)
Who does what...?

Division of labor
Specialization of work
Automatic task / goods allocation
Data / Object clustering
SOCIAL DECISION-MAKING, CONSENSUS

Leaders / Followers

Quorum-based responses

Distributed consensus

Consensus is reached when:

\[ x_i(n) = \frac{1}{N} \sum_{j=1}^{N} x_j(0), \forall i \]

n: Time index
N: number of nodes
Social Decision-making, Consensus

Social choice
Voting theory

---

**Social Choice**

**Voting Theory**

**Arrow's Impossibility Theorem**

No voting system can satisfy all four fairness criteria in all cases.

<table>
<thead>
<tr>
<th>Number of Voters</th>
<th>10</th>
<th>10</th>
<th>10</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>1st Choice</td>
<td>A</td>
<td>B</td>
<td>C</td>
<td>D</td>
</tr>
<tr>
<td>2nd Choice</td>
<td>B</td>
<td>C</td>
<td>D</td>
<td>A</td>
</tr>
<tr>
<td>3rd Choice</td>
<td>C</td>
<td>D</td>
<td>A</td>
<td>B</td>
</tr>
<tr>
<td>4th Choice</td>
<td>D</td>
<td>A</td>
<td>B</td>
<td>C</td>
</tr>
</tbody>
</table>

- No candidate has a majority.
- No Condorcet Winner.
- In fact regardless of the candidate, ¾ of the voters prefer someone else.
- This is called Condorcet's Voting Paradox.

**Condorcet Paradox:**

Cyclic majority:

- Preference: A > B > C > A → A > B > C
- A > B > C > A
- 14.7
- 15.8
- 13.8
- Weakest link

**Method of Condorcet (1743–94)**

**Essay sur l'application... (1785)**

Condorcet paradox: Cyclic majority

- A > B > C > A
- 14.7
- 15.8
- 13.8
Game Theory

Conflict (and cooperation) in multi-agent systems
Equilibrium concepts: minimax, Nash, correlated, leader-follower

J. von Neumann (a pure genius)

R. Aumann (2005 Nobel)

H. von Stackelberg

J. Nash (1994 Nobel)
Conflict (and cooperation) in multi-agent systems: Evolutionary games, social welfare, optimization concepts

J. Maynard Smith

V. Pareto
Mobile agents in crowded and/or highly constrained scenarios

Emergence of order: flows and structures

Flocking
Topology / Formation maintenance
NEURAL MODELS

- Backfed Input Cell
- Input Cell
- Noisy Input Cell
- Hidden Cell
- Probabilistic Hidden Cell
- Spiking Hidden Cell
- Output Cell
- Match Input Output Cell
- Recurrent Cell
- Memory Cell
- Different Memory Cell
- Kernel
- Convolution or Pool

- Perceptron (P)
- Feed Forward (FF)
- Radial Basis Network (RBF)
- Recurrent Neural Network (RNN)
- Long / Short Term Memory (LSTM)
- Gated Recurrent Unit (GRU)
- Auto Encoder (AE)
- Variational AE (VAE)
- Denoising AE (DAE)
- Sparse AE (SAE)
COURSE ORGANIZATION

- Two main lectures + one recitation lecture that will be used in various ways (to complete main lectures, to make exercises, to read papers together...)

- Weekly (almost) homework: reading and reviewing papers, answering theoretical and conceptual questions, implement models and algorithms and play (i.e., explore) with them

- 55% of grading will be based on homework

- 15% of grading will result from a midterm exam (theory, concepts, math)

- 30% of grading will come from a final project that will focus on at least two of the topics discussed during the course: select and read a few related papers, work out and implement the main ideas but in a (slightly) different scenario, add something new either in the ‘mechanisms’ and/or in the analysis

- The final project can be done in group(s) and will be evaluated based on individual reports and oral presentation

- During lectures, be *alive!* Interact with the instructor, ask questions ...
**COURSE OBJECTIVES**

- Students who successfully complete the course will have acquired **general knowledge of multi-component, complex adaptive systems** in terms of:
  - Challenges related to *modeling, prediction, and control* aspects,
  - *Practice* of effectively implementing such systems.

- By studying (complex) systems related to different domains (engineering, biology, economy, robotics, operations research) the student will be exposed to a truly **interdisciplinary background** of mathematical models and application problems.

- You will acquire foundational knowledge and programming practice in a number of important scientific domains, including:
  - Dynamical systems
  - Complex systems
  - Networks
  - Game theory
  - Social choice
  - Swarm intelligence
  - Distributed decision-making
  - Neural networks