Assignments: 35%
- Students will complete 4/5 assignments based on algorithms presented in class

Lab meets in I1 (West) 109 on Lab Wednesdays
- Lab 0: January 14th (completed)
  - Introduction to Python (No Assignment)
- Lab 1: January 28th
  - Measuring Information (Assignment 1)
  - Graded
- Lab 2: February 11th
  - L-Systems (Assignment 2)
  - Graded
- Lab 3: March 25th
  - Cellular Automata & Boolean Networks (Assignment 3)
  - Graded
- Lab 4: April 8th
  - Genetic Algorithms (Assignment 4)
    - Due: April 22nd
- Lab 5: April 22nd
  - Ant Clustering Algorithm (Assignment 5)
    - Due May 4th
Readings until now

- Class Book
    - Chapters 1, 2, 3, 7, 8
    - Chapter 5, all sections
    - Section 7.7, 8.3.1, 8.3.6, 8.3.8-10

- Lecture notes
  - Chapter 1: “What is Life?”
  - Chapter 2: “The Logical Mechanisms of Life”
  - Chapter 3: “Formalizing and Modeling the World”
  - Chapter 4: “Self-Organization and Emergent Complex Behavior”
  - Chapter 5: “Reality is Stranger than Fiction”
    - posted online @ http://informatics.indiana.edu/rocha/i-bic
Projects

- Due by May 6th in Oncourse

  - ALIFE 15 (14)
    - Actual conference due date: 2016
    - http://blogs.cornell.edu/alife14nyc/
      - 8 pages (LNCS proceedings format)
    - http://www.springer.com/computer/lncs?SGWID=0-164-6-793341-0

- Preliminary ideas overdue!

- Individual or group
  - With very definite tasks assigned per member of group
ant clustering algorithm (ACA)

based on dead body cleaning

- Very simple rules for colony clean up
  - **Pick dead ant.** If a dead ant is found pick it up (with probability inversely proportional to the quantity of dead ants in vicinity) and wander.
  - **Drop dead ant.** If dead ants are found, drop ant (with probability proportional to the quantity of dead ants in vicinity) and wander.

Data vector: $X$

ant clustering algorithm (ACA) for multivariate data

- Group n-dimensional data samples in 2-dimensional grid

**Data vector: \( X_1 \)**

<table>
<thead>
<tr>
<th>( x_{1,1} )</th>
<th>( x_{1,2} )</th>
<th>( x_{1,3} )</th>
<th>...</th>
<th>( x_{1,n-1} )</th>
<th>( x_{1,n} )</th>
</tr>
</thead>
</table>

**Data vector: \( X_2 \)**

<table>
<thead>
<tr>
<th>( x_{2,1} )</th>
<th>( x_{2,2} )</th>
<th>( x_{2,3} )</th>
<th>...</th>
<th>( x_{2,n-1} )</th>
<th>( x_{2,n} )</th>
</tr>
</thead>
</table>

Distance between two data samples (in original space):

\[
D(x_i, x_j) = \sqrt{\sum_{k=1}^{n} (x_{i,k} - x_{j,k})^2}
\]

e.g. Euclidean

Ants see data points in a certain neighborhood

\( s^2 \): area of neighborhood (side \( s \), radius 1)
Ant clustering algorithm (ACA) using thresholds

Clustering rules
- **Pick data sample**
  If there are few similar
- **Drop data sample**
  If there are many similar

Probability of picking up
$$p_p(x_i) = \left( \frac{k_1}{k_1 + f(x_i)} \right)^2$$

Probability of dropping
$$p_d(x_i) = \left( \frac{f(x_i)}{k_2 + f(x_i)} \right)^2$$

Neighborhood Similarity or density measure
$$f(x_i) = \begin{cases} \frac{1}{s^2} \sum_{x_j \in \text{Neigh}(s \times s)} \left( 1 - \frac{D(x_i, x_j)}{\alpha} \right) & \text{if } f > 0 \\ 0 & \text{otherwise} \end{cases}$$

Improved with
- Different moving speeds, Short-term memory, Behavioral switches
- Cooling cycle for thresholds, progressive vision, pheromone reinforcement

Reduces dimensionality
No a priori number of clusters
Overshoots number of clusters
ant clustering algorithm (ACA)

The workings

1. Project high-dimensional data items onto 2-dimensional grid randomly
2. Distribute $N$ ants randomly on grid
3. repeat
   - For every ant $i$ in colony
     - Compute neighborhood density $f(x_i)$
     - If ant $i$ is unloaded and its cell is occupied with data item $x_i$ then pick up $x_i$ with probability $p_p(x_i)$
     - Else if ant $i$ is loaded with $x_i$ and its cell is empty drop $x_i$ with probability $p_d(x_i)$
     - Move randomly to neighbor cell with no ant
4. Until maximum iterations
Inspired by brood sorting

Same principle as Clustering

Rules

- **Pick data sample of type** \( t \)
  
  If there are few of type \( t \)

- **Drop data sample of type** \( t \).
  
  If there are many of type \( t \)

\[
\begin{align*}
 p_p (x_i | t) &= \left( \frac{k_1}{k_1 + f_t(x_i)} \right)^2 \\
 p_d (x_i | t) &= \left( \frac{f_t(x_i)}{k_2 + f_t(x_i)} \right)^2
\end{align*}
\]

Probability of picking up item of type \( t \)

Probability of dropping item of type \( t \)

\[
f_t(x_i) = \begin{cases} 
\frac{1}{s^2} \sum_{x_j \in \text{Neigh}_t(s \times s)} \left( 1 - \frac{D(x_i, x_j)}{\alpha} \right) & \text{if } f > 0 \\
0 & \text{otherwise}
\end{cases}
\]

Neighborhood density of type \( t \)
based on ant algorithm


Bristol Robotics Laboratory.

Path optimization

- Stigmergy
  - **Reinforcement**: Shortest path contains probabilistically more pheromone
    - First ants to get to food source are those using the shortest path, so pheromone remains stronger in the whole path, which makes them choose the path more often when going back

- Dependence on dynamic parameters (self-organization)
  - Pheromone evaporation, number of ants, length of paths
    - If shortest path is introduced much later, it will not be chosen unless pheromone evaporates very quickly

- Pheromone release is proportional to food source quality
  - **Exploitation** of better sources

- Ants wander off path with a certain probability
  - Random behavior necessary for exploration of space

- Distributed search
  - Population of foraging ants

- Collective Path Optimization (global coordination)
  - A single ant (one solution) cannot solve it, path optimization is a property of the collective
robustness

pheromone evaporation

E. Bonabeau
**basic definitions**

*Directed graph*
- connected

*Undirected graph*
- disconnected

**Vertices**
- \( V \)

**Edges**
- \( E \)

**Path**: Sequence of vertex, edge, vertex, edge, etc.

**Weighted Graphs**
- \( w(e) \in R \)

**Adjacency Matrix**

<table>
<thead>
<tr>
<th>( V_i )</th>
<th>1</th>
<th>0.3</th>
<th>0.5</th>
<th>0.2</th>
<th>0.2</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.8</td>
<td>1</td>
<td>0.3</td>
<td>1.9</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>0.4</td>
<td>0.5</td>
<td>1</td>
<td>0.7</td>
<td>0.9</td>
<td></td>
</tr>
<tr>
<td>0.1</td>
<td>0.5</td>
<td>0.6</td>
<td>1</td>
<td>0.7</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>0.9</td>
<td>0.4</td>
<td>0.6</td>
<td>1</td>
<td></td>
</tr>
</tbody>
</table>

**Example Adjacency Matrix**
- \( w_{i,j} = 0.6 \)
finding the shortest path

- Start with a weighted graph where edge weights are distances \( d(e) \).
- A solution is a path from vertex \( s \) to vertex \( d \)
  - Length of path \( p \) is \( \sum_{e \in p} d(e) \)
- Pheromone level on edge \( e_{i,j} : \tau_{i,j} \)
- Pheromone evaporates
  - \( \tau_{i,j}(t+1) = (1-\rho) \tau_{i,j}(t) \)
- Population of artificial ants
  - Ant \( z \) traverses a edge (or path) at each iteration \( t \)
  - Releases pheromone every time it traverses an edge: \( \Delta \tau \)
  - Chooses next path to traverse after reaching vertex \( v_i \):

\[
P_{i,j}^z = \frac{\left(\tau_{i,j}\right)^a (d_{i,j})^{-b}}{\sum_{k \in N_i} \left(\tau_{i,k}\right)^a (d_{i,k})^{-b}}
\]

Desirability or visibility
Traveling-sales ants

- $d_{ij}$ = distance between city $i$ and city $j$
- $\tau_{ij}$ = virtual pheromone on edge $(i,j)$
- $m$ agents, each building a tour
- At each step of a tour, the probability to go from city $i$ to city $j$ is proportional to $\left(\tau_{ij}\right)^a \left(d_{ij}\right)^{-b}$
- After building a tour of length $L$, each agent reinforces the edges is has used by an amount proportional to $1/L$
- The virtual pheromone evaporates: $\tau \rightarrow (1-\rho) \tau$

E. Bonabeau
ant colony optimization (ACO)

For the traveling salesman problem

- Pheromone release proportional to quality of solution

\[ \Delta \tau_{i,j}^z = \frac{C}{L^z} \]

Length of path completed by ant \( z \)

\[ p_{i,j}^z = \frac{(\tau_{i,j})^a (d_{i,j})^{-b}}{\sum_{k \in N_i} (\tau_{i,k})^a (d_{i,k})^{-b}} \]
Multiple solutions

Next lectures

Class Book
  - Chapter 5, all sections
  - Section 7.7, 8.3.1, 8.3.6, 8.3.8-10

Lecture notes
- Chapter 1: “What is Life?”
- Chapter 2: “The logical Mechanisms of Life”
- Chapter 3: Formalizing and Modeling the World
- Chapter 4: “Self-Organization and Emergent Complex Behavior”
- Chapter 5: “Reality is Stranger than Fiction”
  - posted online @ http://informatics.indiana.edu/rocha/i-bic

Optional materials
- *Scientific American*: Special Issue on *the evolution of Evolution*, January 2009.