Computation in Cellular Automata
Lab Assignments: 35% (ISE-483), 25% (SSIE-583)
- Complete 5 (best 4 graded) assignments based on algorithms presented in class
  - Lab 3: Cellular Automata and Boolean Networks (Assignment 3)
    - March 20th
    - Due March 31st

SSIE – 583 -Presentation and Discussion: 25%
- Present and lead the discussion of an article related to the class materials
  - Enginet students post/send video or join by Zoom
- Next Presentation March 13th
  - TBA
Class Book

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
  - posted online @ http://informatics.indiana.edu/rocha/i-bic

Papers and other materials
- Criticality and Self-Organization
Projects

- Due by May 8th in Brightspace, “Final Project Paper” assignment
  - ALIFE 2023
    - Not to submit to actual conference due date (March 13th)
    - [https://2023.alife.org/](https://2023.alife.org/)
    - 8 pages, author guidelines:
      - /calls/call-for-papers-extended-abstracts
      - MS Word and Latex/Overleaf templates
  - Preliminary ideas **by March 27**
    - Submit to “Project Idea” assignment in Brightspace.
- Individual or group
  - With very definite tasks assigned per member of group

---

**ALIFE 2023**

Tackle a real problem using bio-inspired algorithms, such as those used in the labs.
quorum sensing or what decision to take? (Density Classification)

Imagine automata as agents

\[ K^{|N|} = 2^7 = 128 \]

Each cell only has access to local information

Local neighbourhood (LNC) contains seven cells, two allowed states (red or white) -> \(2^7\) possible LNCs

What colour is in the majority?

A possible strategy, out of \(2^{2^7}\) possible strategies...
density classification task

Each cell only has access to LOCAL information.

local neighbourhood (LNC) contains seven cells, two allowed states (red or white) -> \(2^7\) possible LNCs.

A possible strategy, out of \(2^7\) possible strategies...

What colour is in the majority?

Typically chaotic behavior
No convergence

\[ K^{\left\lfloor N \right\rfloor} = 2^7 = 128 \]

\[ P = 0 \]
density classification task

\[ K \left\lfloor \frac{N}{K} \right\rfloor = 2^7 = 128 \]

local strategy: majority rule

Each cell only has access to LOCAL information

local neighbourhood (LNC) contains seven cells
two allowed states (red or white) \( \rightarrow 2^7 \) possible LNCs

What colour is in the majority?

\[ P = 0 \]

Isolated groups
No information transmission
density classification task

$$K^{|N|} = 2^7 = 128$$

$$P \in [53\%, 60\%]$$

Each cell only has access to LOCAL information.

local neighbourhood (LNC) contains seven cells two allowed states (red or white) \(\Rightarrow 2^7\) possible LNCs

What colour is in the majority?

A possible strategy, out of \(2^7\) possible strategies...

“blind” spreading of local information
No information integration
Not much better than random choice

GA to evolve rules for DCT [1994]
density classification task

\[ K^{N} = 2^7 = 128 \]

Integration and transmission of information across population

Some other rules that are capable of collective information processing over time and space can solve this task with a range of performances where \( P_{149}^{10^5} > 80\% \)

Local neighbourhood (LNC) contains seven cells, two allowed states (red or white), \( \Rightarrow 2^7 \) possible LNCs.

What colour is in the majority?

Emergent computation strategies
Integration and transmission of information across population

Some other rules that are capable of collective information processing over time and space can solve this task with a range of performances where $P_{14^5} > 80\%$
collective (emergent) computation via computational mechanics

How to characterize complex behavior?

GA to evolve rules for DCT [1994]

- Particle interaction scheme
  - Rules like a production grammar
  - the presence (collision) of two particles produces other particles
  - Transfer information across the lattice
  - Loci of information processing
  - integrate local information globally to solve the nontrivial density task
  - Higher performance than block expansion

Crutchfield & Mitchell [1995]. *PNAS* 92: 10742-10746


<table>
<thead>
<tr>
<th>Regular Domains</th>
<th>( \Lambda^0 - {0+} ), ( \Lambda^1 - {1+} ), ( \Lambda^2 = {01+} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Particles (velocities)</td>
<td>( \alpha = \Lambda^0 \Lambda^1 (-) ), ( \beta = \Lambda^1 \Lambda^0 (0) ), ( \gamma = \Lambda^0 \Lambda^2 (-1) ), ( \mu = \Lambda^2 \Lambda^1 (1) ), ( \delta = \Lambda^2 \Lambda^0 (-3) ), ( \eta = \Lambda^1 \Lambda^2 (3) )</td>
</tr>
<tr>
<td>Observed Interactions</td>
<td>decay: ( \alpha \cdot \gamma + \mu )</td>
</tr>
<tr>
<td></td>
<td>react: ( \beta + \gamma - \eta, \mu + \beta - \delta, \eta + \delta - \beta )</td>
</tr>
<tr>
<td></td>
<td>annihilate: ( \eta + \mu - \Lambda^1, \gamma + \delta - \Lambda^0 )</td>
</tr>
</tbody>
</table>


How do best rules solve the problem?

comparison of different automata

GKL Rule
- 3 domains
- 6 particles

GP Rule
- > 10 domains
- > 90 particles

\[ p = \frac{d!}{(d-2)!} \]
quantifying redundancy (canalization) in automata nodes

nonlinear measure of effective connectivity

\[ k(x_4) = 3 \]

\[ x_4 = x_1 \land x_2 \]

\[ k_e(x_4) = 1.25 \]

Prime Implicants (Quine-McCluskey)

minimal transition control: set of wildcard schemata is DNF of prime implicants (Blake Canonical Form)

Measuring redundancy and its dual effectiveness

input redundancy:

\[ k_r(x) = \text{mean number of "#" in LUT} \]

\[ k_r(x_4) = \frac{1 \times 2 + 2 \times 6}{8} = 1.75 \]

effective connectivity:

\[ k_e(x) = k(x) - k_r(x) \]

edge effectiveness

\[ k_e(x_i) = \sum_{j=1}^{k} e_{ji} \]

\[ e_{ji} = 1 - r_{ji} \]

quantifying redundancy (canalization) in automata nodes

nonlinear measure of effective connectivity

Measuring redundancy and its dual effectiveness

input redundancy:

\[ k_r(x) = \text{mean number of "#" in LUT} \]

\[ k_r(x_4) = \frac{1 \times 2 + 2 \times 6}{8} = 1.75 \]

effective connectivity:

\[ k_e(x) = k(x) - k_r(x) \]

\[ k_e(x_3) = \sum_{j=1}^{k} e_{ji} \]

edge effectiveness

\[ r_{ji} = \frac{\sum_{f_\alpha \in F_i} \text{avg} \ (j \rightarrow \#)_\alpha}{|F_i|}, \]

\[ e_{ji} = 1 - r_{ji}. \]

How do best rules solve the problem?

Comparison of different automata

How do interactions at the micro-level generate desired macro-level behavior?

How to control/program behavior?

How to compare and classify behavior?

GKL Rule
- 3 domains
- 6 particles

GP Rule
- > 10 domains
- > 90 particles!!!

\[ p = \frac{d!}{(d-2)!} \]
It takes redundancy to solve

Ignore most input information

Solving by schemata

- Each automaton ignores most inputs

GKL Rule
- 3 domains
- 6 particles

GP Rule
- > 10 domains
- > 90 particles!!!

search in redescription (canalization) space

canalization (redundancy) improves evolutionary search

- Created much smoother search space
  - Allows more focused search of rules
    - Canalization, neutrality, robustness?
  - Second best rule in 1-D CA (best-known PS rule)
    - Best split-performance
  - Best rule in 2-D CA
- reason about emergent computation in new ways
  - Process-symmetry

Studying emergence

linking local and global/collective behavior

- Are emergent patterns good for explanation?
  - Do stripes or spots explain the “system”?
- Canalization (dynamical redundancy) is a powerful idea
  - Capture loci of control and building blocks of information transmission

---

**GKL Rule**
- 3 domains
- 6 particles

**GP Rule**
- > 10 domains
- > 90 particles!!!

---

<table>
<thead>
<tr>
<th>Rule</th>
<th>Hexadecimal Representation</th>
<th>(P_{149})</th>
<th>Produced by</th>
<th>Source</th>
</tr>
</thead>
<tbody>
<tr>
<td>(\Phi_{GKL})</td>
<td>5000500050050505f5005ff5f</td>
<td>0.8143</td>
<td>HE</td>
<td>Gaca et al., 1978</td>
</tr>
<tr>
<td>(\Phi_{GP1995})</td>
<td>50005005050505055ff5ff5f5f</td>
<td>0.8212</td>
<td>GP</td>
<td>Andre et al., 1996</td>
</tr>
</tbody>
</table>

Are emergent patterns good for explanation?
- Do stripes or spots explain the “system”?

Canalization (dynamical redundancy) is a powerful idea
- Capture loci of control and building blocks of information transmission

(GKL Rule)
- 3 domains
- 6 particles

(GP Rule)
- > 10 domains
- > 90 particles!!!

(mechanistic) reductionism vs emergence:
what is the best explanation?

the game of life

<table>
<thead>
<tr>
<th>Sum $N^8$</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
</tr>
</thead>
<tbody>
<tr>
<td>$x_{i,i} = 0$</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>$x_{i,i} = 1$</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

1) Any living cell with fewer than two neighbors dies of loneliness.
2) Any living cell with more than three neighbors dies of crowding.
3) Any dead cell with exactly three neighbors comes to life.
4) Any living cell with two or three neighbors lives, unchanged, to the next generation

Introduced in Martin Gardner’s *Scientific American* “Mathematical Games” Column in 1970.

Conway was interested in a rule that for certain initial conditions would produce patterns that grow without limit, and some others that fade or get stable.

Popularized CAs.
Simple Attractors

*Block*

Blinkers

More complicated attractors
moving patterns

glider
unbounded growth

a threshold of complexity?

runs 1103 steps before settling down into 6 gliders, 8 blocks, 4 blinkers, 4 beehives, 1 boat, 1 ship, and 1 loaf.

something that lasts forever?

R-pentomino
Unbounded growth but not complexity

Fires a glider every 30 iterations.
unbounded complexity requires information

1) Patterns that can implement information, descriptions, and construction
2) Gliders, guns, blocks, eaters

Very brittle
Built, not evolved
Not evolving

Universal Turing Machine on game of life!!!
Rule 110

- Radius 1
  - Neighborhood = 3
- Binary
  - $2^3 = 8$ input neighborhoods
  - $2^8 = 256$ rules

Information in attractor patterns

http://mathworld.wolfram.com/Rule110.html
Universal Computation
- Identification of gliders, spaceships, and other long-range or self-perpetuating patterns
  - On the background domain produced by rule 110
    - 14 cells repeat every seven iterations: 00010011011111
- Collisions and combinations of glider patterns are exploited for computation.
is self-organization enough

- Systems biology models operate in near critical regime, though many are ordered.
- Dynamical systems capable of computation exist before the edge of chaos.
  - A wider transition due to redundancy?
- Most important information transmission and computation in Biology an altogether different process than self-organization.
  - Turing/Von Neumann memory.
readings

- **Class Book**
    - Chapter 2.

- **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
    - posted online @ http://informatics.indiana.edu/rocha/i-bic

- **Papers and other materials**
  - **Optional**
      - Chapter 2, all sections
      - Chapter 7, sections 7.3 – Cellular Automata
      - Chapter 8, sections 8.1, 8.2, 8.3.10
      - Chapters 10, 11, 14 – Dynamics, Attractors and chaos