Evolving Cellular Automata to Perform Computations: Mechanisms and Impediments

Melanie Mitchell, James Crutchfield, and Peter Hraber

Presented by:
Geoff Alexander, Xu Zhang, and Lengge Se

February 18, 2013

Overview

1. The main idea of GA evolves CA to perform particular computational task
2. Details of the GA and CAs
3. GA Impediments
4. Discussion

Introduction

Using GA to evolve cellular automata to perform a particular computational task – one dimensional density classification

Why CA

No central unit to do counting

Just share information with neighbor

Different from standard approach to parallel computation

CA Review and Terminology

A cell's state at time t+1 is depended on its own state and the states of some neighboring cells at time t.

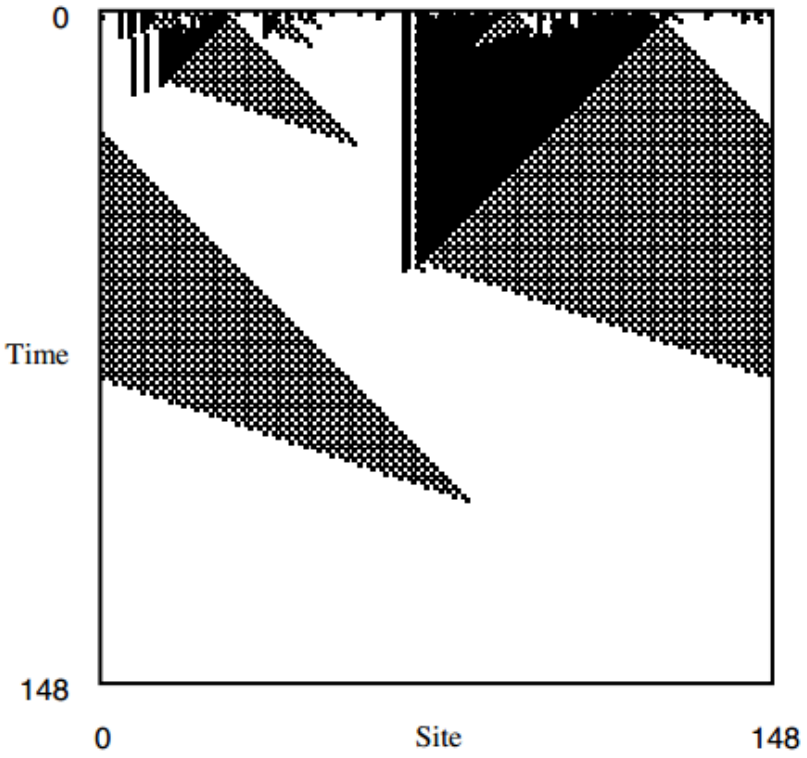
- N cells(N = 149), k states (k = 2), state s {0,1}, r neighbors on both sides
- $\rho(s)$ density of 1s in a configuration s
- $\Phi(s_t)$ global mapping

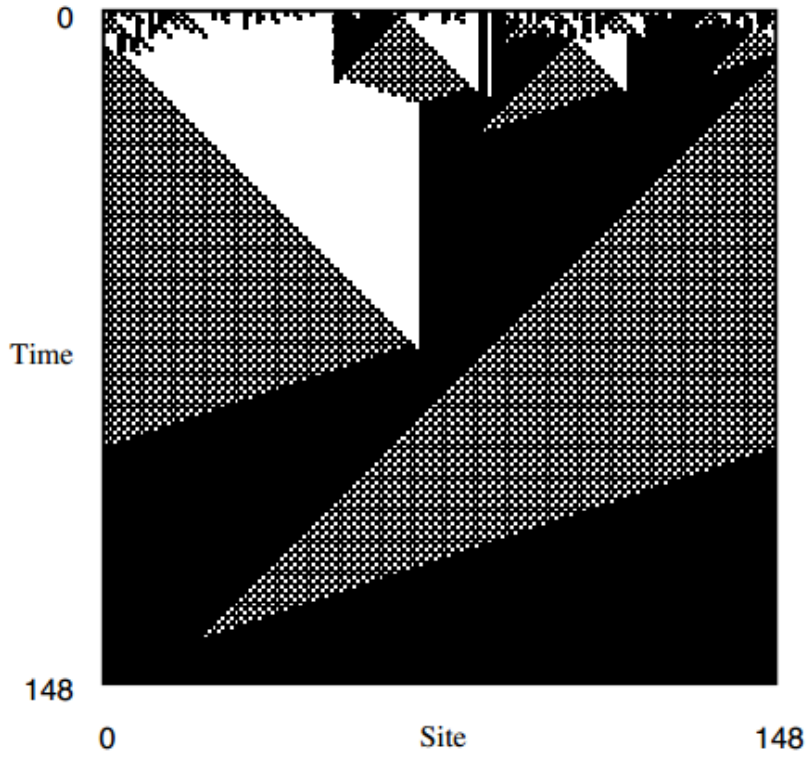
Computational Task

$$\mathbf{T}_{\rho_c}(N,M) = \left\{ \begin{array}{ll} \Phi_M(\mathbf{s}_0) = 0^N & \text{if } \rho(\mathbf{s}_0) < \rho_c \\ \Phi_M(\mathbf{s}_0) = 1^N & \text{if } \rho(\mathbf{s}_0) > \rho_c \\ \text{undefined} & \text{if } \rho(\mathbf{s}_0) = \rho_c \end{array} \right\} \forall \mathbf{s}_0 \in \{0,1\}^N$$

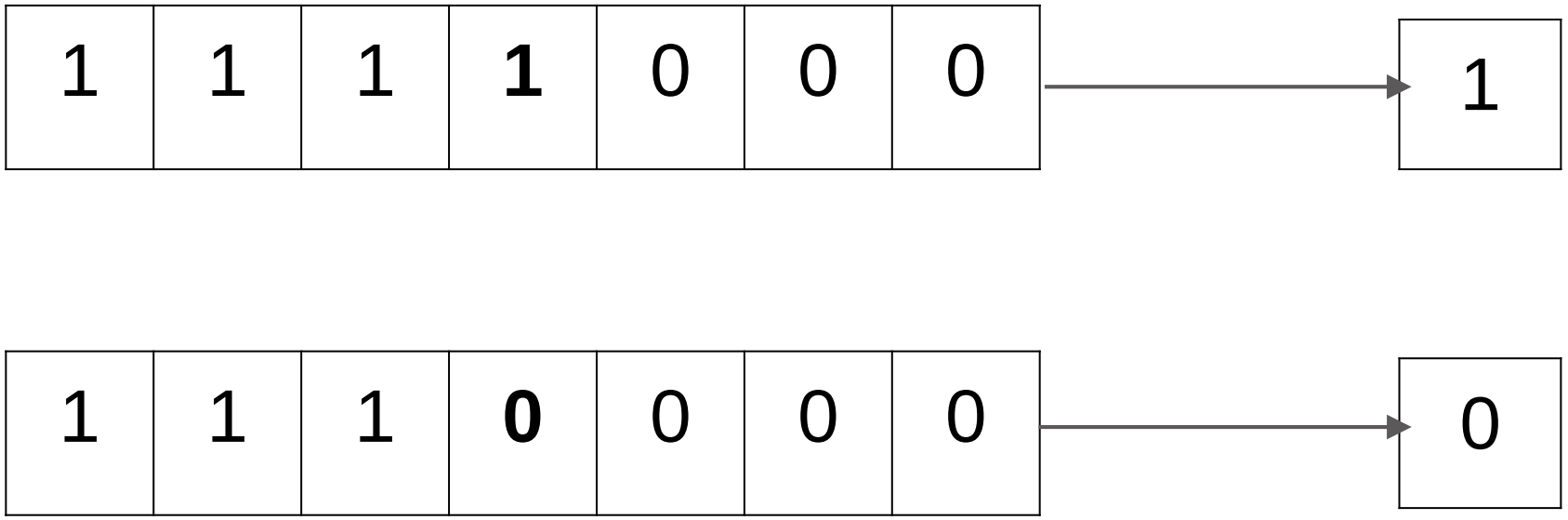
GKL rule

$$s_{t+1}^i = \phi(\eta_t^i) = \begin{cases} \text{majority}[s_t^i, s_t^{i-1}, s_t^{i-3}] & \text{if } s_t^i = 0 \\ \text{majority}[s_t^i, s_t^{i+1}, s_t^{i+3}] & \text{if } s_t^i = 1 \end{cases}$$

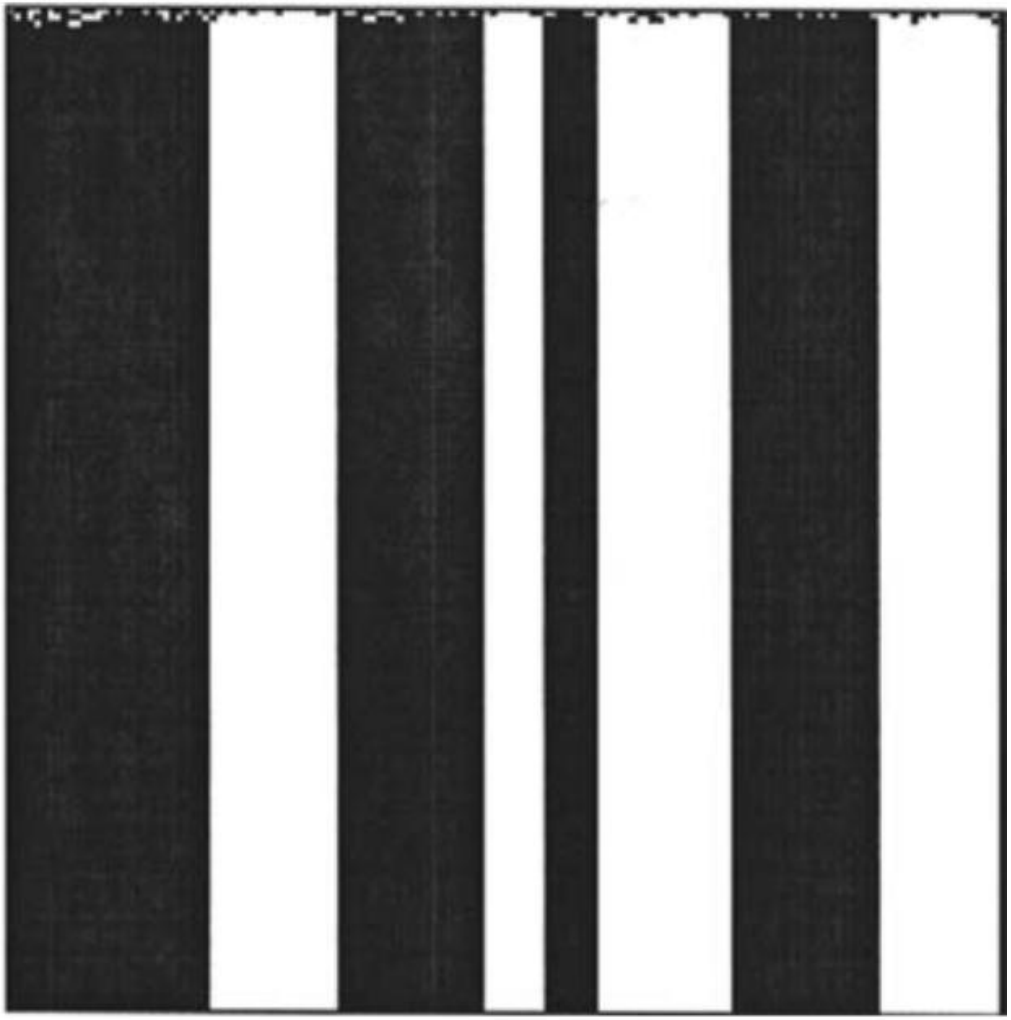




How "majority rule" works



Space & time behavior of "local majority vote" cellular automaton



The fitness of a rule

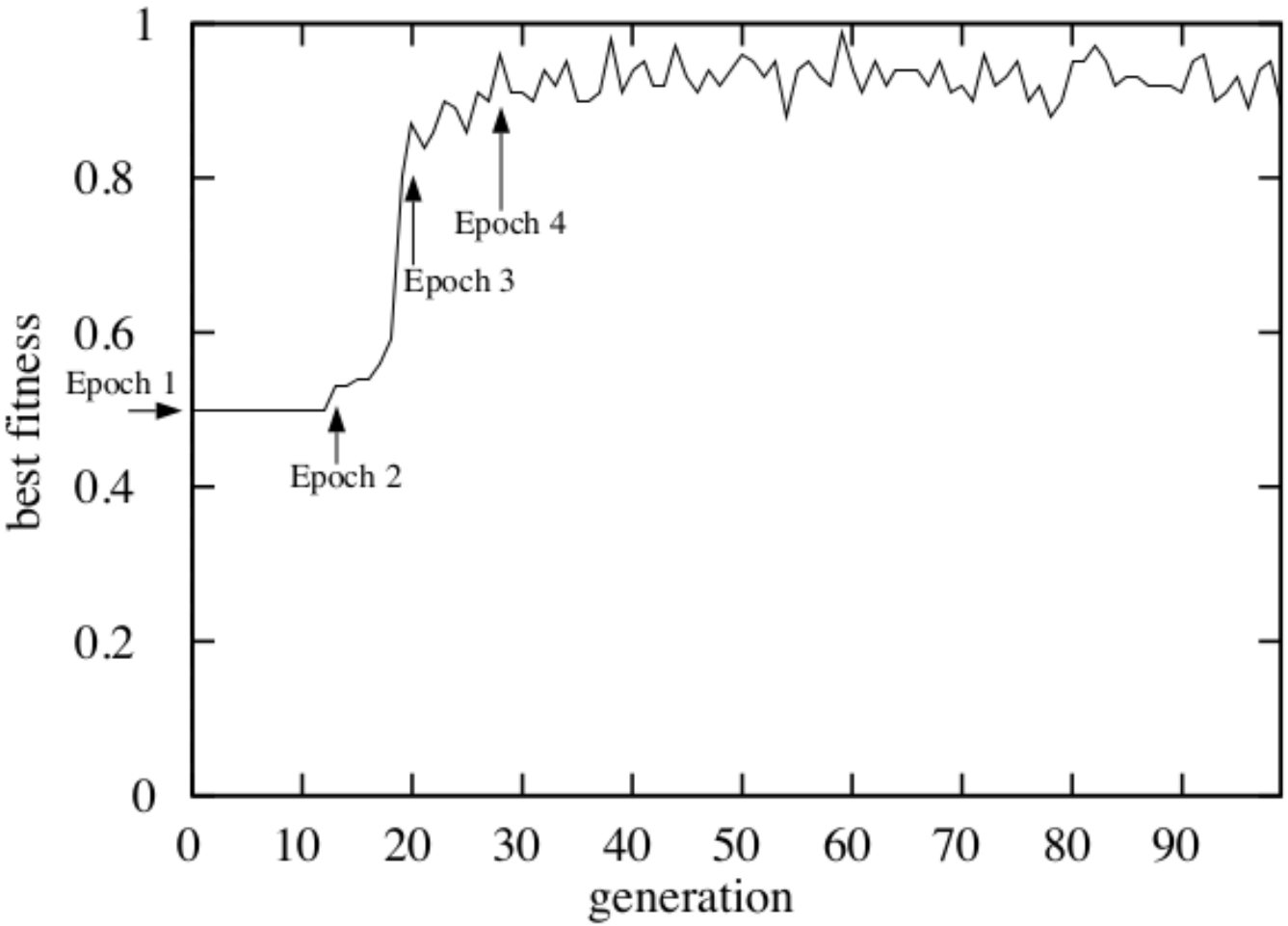
1. Randomly choose 100 ICs. (50 majority 1s, 50 majority 0s)
2. Run this rule on each ICs. (M time steps or reach a fixed point)
3. Determine if final pattern is right for each IC.

* This rule's fitness is the # of correct final patterns / 100.

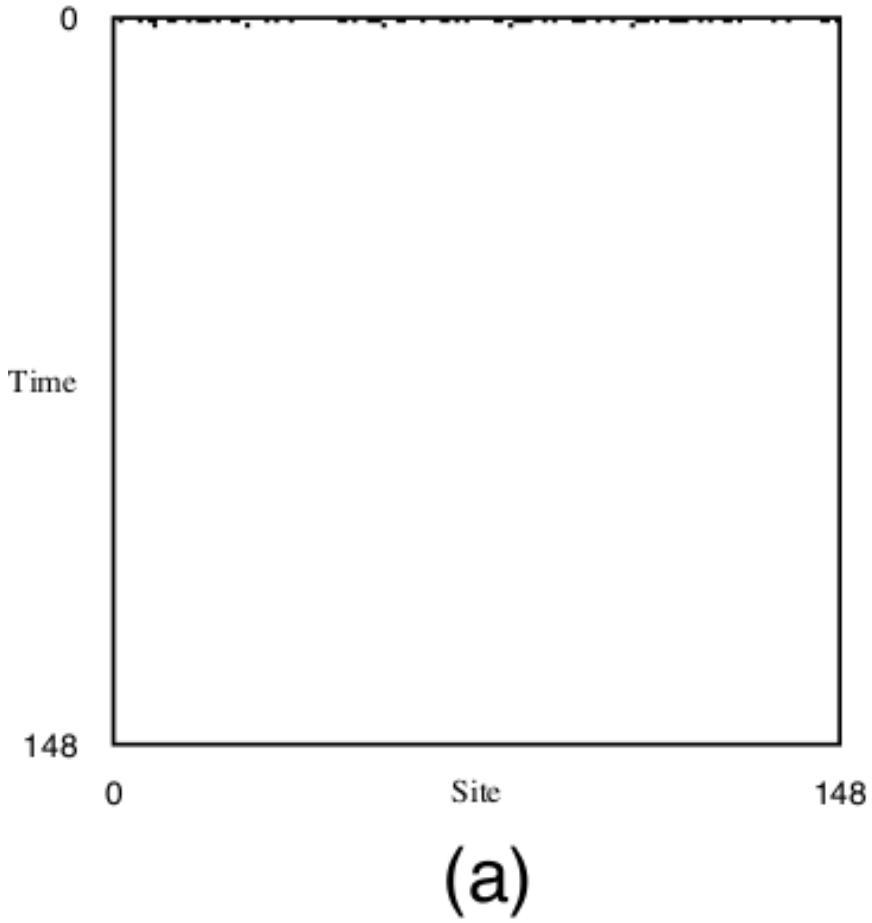
How GA works?

1. Choose P rules (uniformly distributed [0,1]).
2. Each rule's fitness is calculated
3. Population P is ranked in order of fitness
4. Some highest fitness rules E is copied without modification to next generation
5. The remaining P-E rules formed by crossovers.

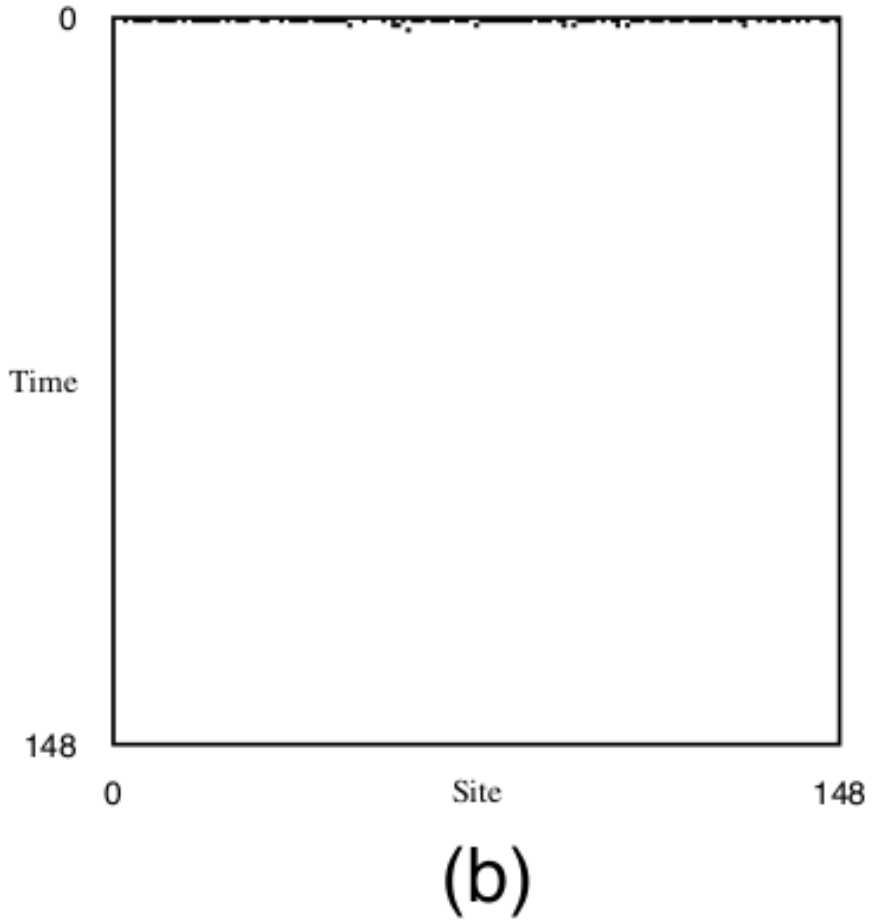
Best fitness versus generation



Epoch 1: Best rules specialize on low or high ρ_0

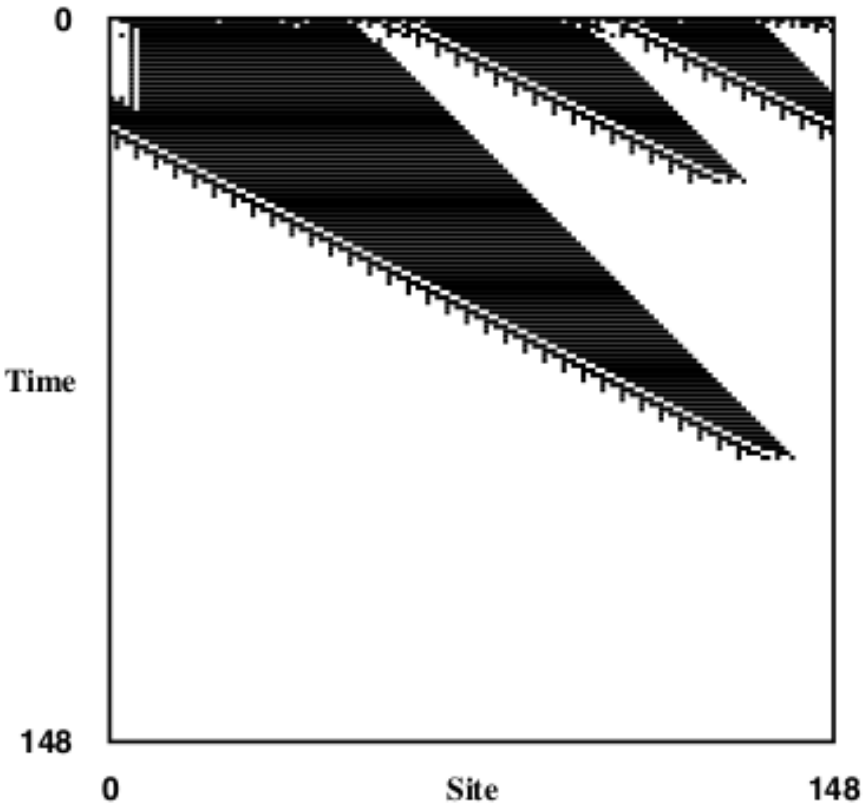


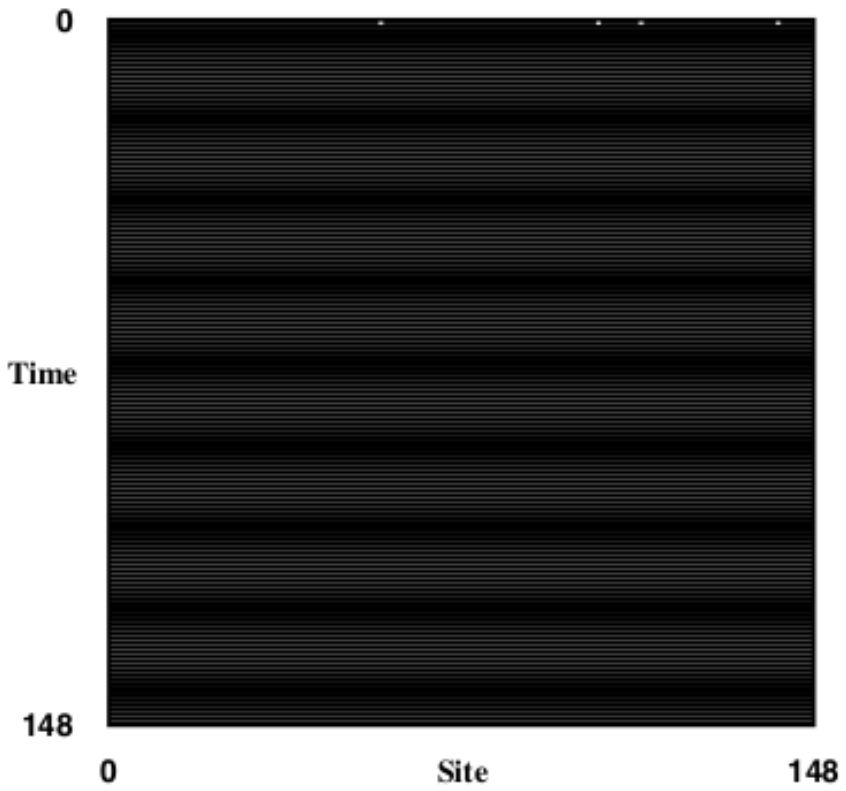
(a)



(b)

Epoch 2: Best rules correctly classify additional "extreme" ρ_0





GA Impediments

- Evolved rules still misclassify densities near 1/2
  - Is this an inherent property of GA's when applied to density classification?
  - Inherent property of problem representation?
- What keeps the GA from finding a perfect rule set?

GA Impediments

Symmetry Breaking

- GA tends to produce specialist rules
- GA prefers rules that give short-term gains at expense of long-term improvements
  - Specialist rules provide simple way to improve fitness score
- Hypothesized that this is general feature of GAs and natural evolution in general.

GA Impediments

Loss of Diversity

- New strategies result in large fitness gains
- Strategies quickly spread through population
- Helps GA improve in early stages but prevents improvements later on

Summary

- Using GA to evolve rules for CA
- GA Overview
  - GA Implementation
  - Different epochs
- GA Impediments
  - Symmetry Breaking
  - Loss of Diversity

Discussion

- Is it possible to to evolve rule(s) that will always correctly solve density classification?
  - Why or why not?
- How can the impediments be worked around or removed?