Computing department

Artificial Life – Ethology

CSc 355

Alan Dix
dixa@comp.lancs.ac.uk

Manolis Sifalakis
mjs@comp.lancs.ac.uk

Lecture Overview

- What is Alife
- Brief history – timeline
- Synthetic Ethology and Food Chains
- Example: Food Chain model
  - Agent Anatomy
  - (Pseudo-) code
  - Sample iteration
  - Results
  - Observations
- Reference List

What is Alife ?

- Alife [Langton] is set of mechanisms used to model and simulate evolving natural systems
  - Insect ecologies, animal behavior, negotiating entities, resource use in artificial economies
  - Studies the evolution of agents, or populations of computer simulated life forms in artificial environments
  - Complements traditional biology by trying to *recreate* biological phenomena

---

What is Alife ?

Alife

Computational Models (soft approach) | Robotics (hard approach) | Biochemistry (wet approach)

Cellular Automata | Neural Networks

- Traditionally AI: a top down approach
- Alife: works from the bottom up

Brief history - Timeline

Jacques de Vaucanson
(1739, pre-computer)
Artificial Duck

↓

John Von Neumann
(late 1940s)
Theory of automata
Self-replicating Machine

↓

Homer Jacobson
(1950s)
Illustrated basic self-replication with a model train set

↓

John Horton Conway
(1960s)
The Game of Life
(Most famous cellular automaton)

↓

Edgar F. Codd
(1968)
Simplified Von Neumann's Cellular automaton from 29 states to 8 states

↓

Christopher Langton
(1979)
First self-replicating computer organism using only an Apple II desktop computer
(1989)
Founder of Artificial Life

↓

Stephen Wolfram
(1982)
1-D Cellular Automata applied to natural phenomena (seashell patterns, plant growth)

↓

The Unit of Theoretical Behavioural Ecology
Free University of Brussels
(mid 1980s)
Self-organization theories to research the behavior of social insects

Synthetic Ethology & Food Chains

- Synthetic Ethology
  - Study of animal behavior in which simple, synthetic organisms are allowed to behave and evolve in a synthetic world.
  - Branch of zoology
- Food Chain
  - Describes the hierarchy of living organisms within an ecosystem.

Carnivore

↓ *consumes*

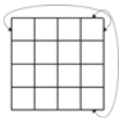
Herbivore

↓ *consumes*

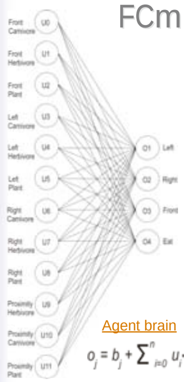
Plant

Example: Food Chain Model (FCm)

- 3 Entities
  - Plants:
    - Fixed location, consumed by herbivores
  - Herbivores:
    - Migratory agents, eat plants, eaten by carnivores
  - Carnivores:
    - Migratory agents, eat herbivores, die from starvation
- Environment
  - Toroid grid
  - Each cell occupied by one or more agents



FCm: Agent Anatomy



Agent brain

$o_j = b_j + \sum_{i=0}^{n} u_i \cdot w_{ij}$

The Agent

Agent perception of the environment

FCm: Agent "Life & Death issues"

- Energy (*E*) / Metabolism
  - Eat → *E = E + 1*
  - For each step → *E = E – X*, (*H: X=1, C X=2*)
  - If *E == 0* → Die
- Reproduction
  - If *E > 90%* → Reproduce asexually
  - Lamarckian: Offspring inherits parents' NNet followed by random mutation of weights
- Death
  - Starvation (no food found)
  - Eaten (only for herbivores

FCm: The (pseudo-) code

```
Main ( )
   Init ( )
   while (run < MAX_RUNS)
      SimulateOnce ( )

Init ( )
   landscape := InitLandscape ( )
   GrowPlants ( landscape [plants] )
   while ( agents < MAX_AGENTS )
      agent := CreateAgent ( )
      if ( agent.type == herbivore )
         PositionAgent ( landscape [herbivores] )
      else
         PositionAgent ( landscape [carnivores] )
```

FCm: The (pseudo-) code

```
SimulateOnce ( )
   forall agent types
      foreach agent
         PerceiveEnvironment (agent)
         ForwardPropagateInputs (agent.Nnet)
         ComputeAction (agent)
      switch (agent.action)
         case TURN_LEFT:
         case TURN_RIGHT:
            agent.direction := UpdateOrientation (agent)
         case MOVE_FRONT:
            agent.position := UpdatePosition (agent)
         case EAT:
            Eat (agent)
```

FCm: The (pseudo-) code

```
...
   UpdateEnergy (agent, agent.action)
   if agent.energy  == 0
      KillAgent (agent)
   else
      agent.age += 1
   if agent.energy > REPRODUCTION_LEVEL
      ReproduceAgent ( agent )
```

FCm: The (pseudo-) code

```
GrowPlants ( )
    location := SelectRandomLocation (landscape [plants] )
    if no plant in location
        landscape [plants] [location.x] [location.y] := 1

CreateAgent ( )
    agent.energy := MAX_ENERGY / 2
    agent.age := 0
    agent.generation := 1
    agent.type := carnivore | herbivore
    foreach neuron in Nnet
        SetWeight (neuron)
```

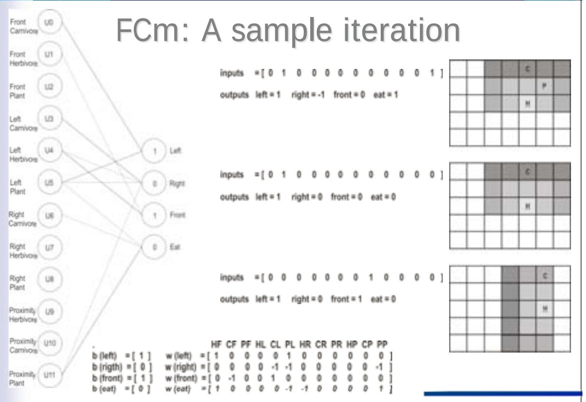
FCm: The (pseudo-) code

```
PositionAgent ( )
    location := SelectRandomLocation (landscape [agent.type])
    if no agent in location
        landscape [agent.type] [location.x] [location.y] := 1
    agent.direction := SelectRandomDirection ( )

Eat ( )
    if agent.type == CARNIVORE
        UpdateLandscape ( landscape [herbivores] )
    else
        UpdateLandscape ( landscape [plants] )
```

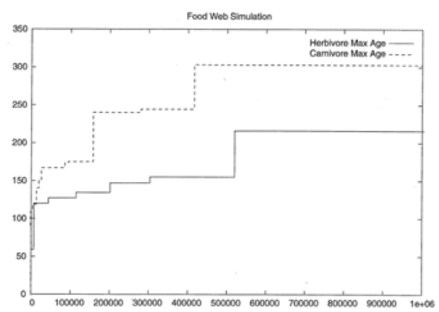
FCm: The (pseudo-) code

```
KillAgent ( )
    UpdateLandscape ( landscape [agent.type] )
    if num of agent of this type < MAX_AGENTS / 4
        CreateAgent ( )

ReproduceAgent ( )
    if num of agents of type < MAX_AGENTS / 2
        // Inheritance of NNet in offspring
        new_agent := DuplicateAgent ( agent )
        // Randomly mutate neuron weights
        foreach neuron in new_agent.Nnet
            if mutation_probability > 50%
                SetWeight (neuron)
        PositionAgent ( landscape [agent.type] )
```

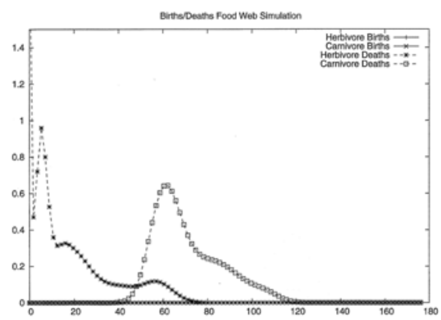
FCm: A sample iteration



FCm: Simulation Results



FCm: Simulation Results



Computing department

FCm: Observations and Conclusions

- Competition
  - Carnivores evolve NNets, good at locating and eating herbivores
  - Herbivores evolve NNets that find plants and avoid carnivores
- Evolved Strategies
  - Herding: Herbivores follow other herbivores in front
  - Ambushing: Carnivores find plants and then wait for herbivores to wander by

Computing department

FCm: Observations and Conclusions

- Parameters tuning
  - Number of plants >= number of herbivores
  - Number of agents must be small so as not to crowd the simulation
  - Number of carnivores <= 2 * Number of herbivores

Computing department

Reference List

- Seminal paper
  - Christopher G. Langton. Artificial Life. Proceedings of interdisciplinary workshop on the Synthesis and Simulation of Living Systems, Los Alamos,1987. Addison-Wesley. 1989
- Zooland: "The Artificial Life Resource"
  - http://surf.de.uu.net/zooland/
- Book chapter on Artificial Life
  - M. Tim Jones. 2003: AI Application Programming. Charles River Media, Inc.

Computing department

Questions ...