GENETIC ALGORITHMS[1]

Microant, Gene, Boid, The John Muir Trail, Prisoner's Dilemma, etc.

How They Are Created and Work

WHAT ARE GENETIC ALGORITHMS?

Genetic algorithms (GA=s) are more complex cellular automata (CA). Typically they operate on a two dimensional CA grid. Both Genetic Algorithms and Cellular Automata are part of a branch of mathematics known as ***Artificial Life*** (or ***A-Life***, or ***Alife***).

Because CA=s and GA=s are based on the same principles they posses all the same attributes. That is, they are open systems, following the dictates of chaos and complexity theory, and operate according to the rules of dissipative structures (positive and negative information/energy feedback, sensitive dependence, local rules/global behavior, and emergent properties.)

GA=s differ from cellular automata in two important ways.

First, whereas the cells in CA=s usually have only two states, alive or dead, cells in genetic algorithms have many states existing simultaneously.

An individual cell, the GA, is assigned a few to many Agenes@, each coding a number of character states (alleles in genetic terminology) which control the attributes and behavior of the GA. The information for the GA is stored in the computer memory, analogously to the way genes are stored on the DNA, in a chromosome, in the nucleus.

Second, this GA package moves around the grid (virtual world, ecosystem) like an individual organism, interacting with the environment and other organisms according to the instructions contained in its genes.

(A CA by contrast has each cell remaining in place, but changing state each generation based on the states of the surrounding cells, and the rules of the CA.)

A more formal definition of a GA is:

> ***Genetic algorithm*** *- a computer program which creates A-Life organisms with electronic genetic information and allows these organisms to exist, interact, and evolve in a computer based ecosystem.*

WHAT ARE GENETIC ALGORITHMS FOR?

Alife systems are theoretical research tools used to try to understand the principles of life by creating artificial life systems and experimenting with them to discover possible mechanisms

[1] Extensive explorations of genetic algorithms can be found in Stephen Levy, 1992, Artificial Life: Pantheon Books, and Stephen Prata, 1993, Artificial Life Playhouse: Waite Group Press (complete with computer disk with artificial life programs, including Life3000).

by which life originated, evolved and behaves. Whether this will be a successful activity has been debated, but we will not go into that here.

Nonetheless, Alife experiments have demonstrated that global order arises spontaneously from the implementation of randomly generated local rules (genetic alleles), randomly mutated, and worked on by natural selection, without need for teleological explanations. That is, we make a distinction between:

☞ **Top-Down Strategies** to explain evolution, which are teleological and require a higher intelligence to design them, and:

☞ **Bottom-Up Strategies** which are non-teleological, and evolve spontaneously out of the working of local rules and natural selection.

GA=s vary widely in their structure, complexity, and purposes, and it is difficult to generalize about them. Many, however, are given names, such as A*Microant,*@ A*Gene,*@ A*Boid,*@ etc. The description below is based on Microant, a relatively sophisticated GA. In the lab we will explore this and some of the other GA's.

MICROANTS are electronic ants that bewhave as dissipative structures in that energy/formation input and output is required to survive. Individual microants contain in their genes information for their own survival and reproduction, nothing more, although their ability to survive varies widely. The characteristics include:

☞ ***Microant has 9 Agenes@*** which code, for example, for vision distance, movement, poison detection, mating. etc. (see last page).

☞ ***The genetic information*** is in binary code, that is, 1's and 0's, somewhat simpler than the 4 nucleic acid base pairs of DNA.

☞ ***The initial genetic make-up*** of the microants is selected at random, just by generating a string of 1's and 0's as long as necessary to define the entire code.
  > Typically a population of dozens of microants is generated randomly.

☞ ***Genes can mutate*** randomly at any rate set by the researcher.
  > Mutations occur just by switching a 1 to a 0, or vice versa.
  > Crossovers and inversions, as with real genes, are also possible.

☞ ***Microants have an energy budget***.
  > Every move, every time step takes energy. If they run out of energy they die.
  > However, every time they land on Afood@ they gain energy, and so can survive a little longer.

☞ ***Microants can fight, share food, and reproduce***, although they have to gather a lot of energy to do it.

Micoants live in a virtual world (the cellular automata grid) where they interact with food, poison, other ants (sometimes of more than one species), and predators (an anteater). This environment is very flexible and can be programmed to have a very wide range of conditions.

If a microant is ***un*** successful for any reason (can=t recognize poison, or food, or predators) it dies, along with its genetic makeup. That is, they are naturally selected out of the gene pool.

If the microant is successful, it and its genotype gathers enough energy to reproduce and create more microants with its genetic makeup. That is, they are naturally selected for.

There are two great advantages to using Microants (and other GA=s) for studying evolution. First, unlike living organisms, the entire genome of the Microant is known, and so as evolution occurs changes in the gene pool can be statistically analyzed to understand exactly what went on during the evolution.

Second, microants live in an environment where all variables are known and controlled (rather than a real, complex ecosystem where many or most of the variables are imprecisely or un- known), and life processes can be stripped down to their essentials (simple ideal models). Because of this, evolutionary experiments can be run over and over, each time with slightly varying environmental or genome conditions to see their effects.

OTHER GENETIC ALGORITHMS

is a simple CA organism with 30 genes, all programed for movement. Each gene has one of four values: up, down, right, left. Net movement of the organism is a combination of 30 movements taken in sequential order. Genes can mutate and do so at random at a preset rate.

There is an energy cost for each move, which can be replenished by landing on a food cell. Reproduction also occurs, as well as random mutation of the genes.

The program starts with Gene GA=s being generated at random. Initially all the GA=s move at random, but as the simulation progresses the population evolves to move in unison, migrating across the screen like marching soldiers.

began with curiosity about what controls schooling behavior in fish and birds. The prevailing opinion was that such behavior required a relatively sophisticated level of intelligence. Boids demonstrated otherwise. With only three rules of behavior . . .

1. Boids need to look around their local area, decide where most other boids are and head in that direction.
2 Boids need to match velocity with their neighbors.
3 Boids need to avoid bumping into each other and obstacles in their path.

. . . the boids exhibit amazingly birdlike behavior. They quickly form a flock and maintain the integrity of the flock as they fly around. When confronted with an obstacle the flock sometimes splits, flies around the obstacle, and reunites on the other side.

In this program the individual boids determine their own direction and perceived center of mass. None of the flocking behavior is explicitly programmed. Rather the flock emerges as a result of the simultaneous interactions between the individual boids, and the principles of local rules/global behavior and emergent properties.