September 18, 1992

To: The 2050 Theoretical Group

From: Joshua Epstein

Re: Using Genetic Algorithms to Grow Artificial Societies

In my August 4 memo to Ed Knapp, I discussed the application of A-Life techniques to the problem of social evolution as a whole: Is there a small set of local rules, that is, rules governing the behavior of individual agents, that over many iterations, will generate a crude caricature of, say, the observed international system--a set of coherent societies with internal structures (e.g., hierarchical, egalitarian) and dynamics, interacting with one another in various cooperative and competitive ways, in an environment that is affected by, and feeds back on, the productive activities of the agents? Core questions might include:

- What system(s) of local rules will generate politically egalitarian societies? Totalitarian societies?
- What rule system(s) will yield social aggregates that are largely peaceful and cooperative? Warlike and competitive?
- Can internal instability--revolutions, the rise and fall of empires--be made to emerge from simple rules?

<u>Proposal:</u> <u>Use Genetic Algorithms to find the rules.</u>

Basically, there are five steps.

<u>Step 1. Target Patterns</u>

First, one needs to define some very simple <u>target</u> patterns/behaviors (Turing test analogues) an artificial social life system should produce as outputs. For instance, can you get emergent social hierarchy after 10^5 iterations?

<u>Step 2. Actual Patterns</u>

Every system, i , of local rules (n-bit <u>strings</u> encoding behavioral rules) generates some social evolution. Check after 10^5 iterations: did hierarchy emerge? There is some actual output that, in principle, one could compare to target output. Specifically, Step 3. Define a Metric on Patterns

Define a mapping ϕ from the set of patterns (target or actual) to the set of real k-vectors (some k). Suppose the target pattern was T and the actual emergent pattern under rule system i was $A(i)$. Then ϕ sends these to k-vectors:

$$\phi : T \rightarrow \phi(T) \in \mathbf{R}^k$$

$$\phi : A(i) \rightarrow \phi[A(i)] \in \mathbf{R}^k$$

With $\| \cdot \|$ The Euclidean norm, define the distance between the target and the emergent actual pattern as

(1) $$d(\phi(T), \phi[A(i)]) = \|\phi(T) - \phi[A(i)]\|$$

Step 4. Define a Fitness on the Set $\{i\}$ of Rule Systems

Given (1) define the fitness of rule system $i \in \{i\}$, call it $F(i)$, as some bounded monotone function of distance $d(\phi(T), \phi[A(i)])$. For example, let

(2) $F(i) = \exp[-d(\phi(T), \phi[A(i)])]$, so $0 < F(i) < 1$. Finally,

Step 5. Let a Genetic Algorithm Search {i}

If we want to know what rule system $i \in \{i\}$ generates the pattern closest to the target, we're simply asking for the string i with highest fitness under (2). So, turn a GA loose on $\{i\}$.

That's the basic idea. Clearly, real problems arise at each step. When I proposed this idea in Michigan (on September 11) to John Holland, Bob Axelrod, and the other BACH group members, the issues of encoding and computational requirements loomed pretty large. How much initial sifting of strings could be done by humans, narrowing the set on which the GA would operate? Is there a simple problem--match some pattern of industrial concentration--that a prototype could handle?

What do you think?