Algorithms red in tooth and claw

Evolutionary computation for design

Simon Greenwold
MIT 10.27.04

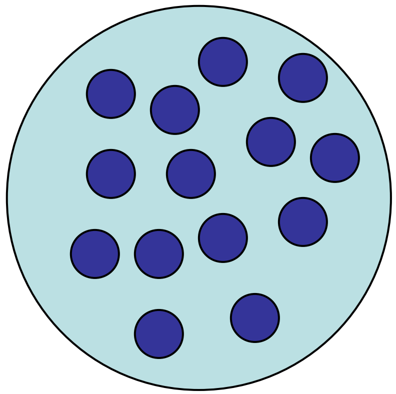
Evolutionary computation: definition

A family of optimization techniques that share

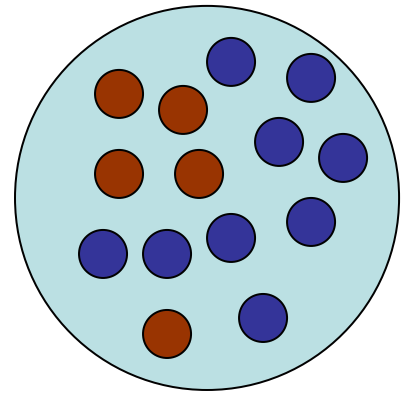
1) Genetic inheritance
2) A fitness metric (objective function)
3) Struggle for survival and reproduction

Nothing here indicates that it has any value for designers

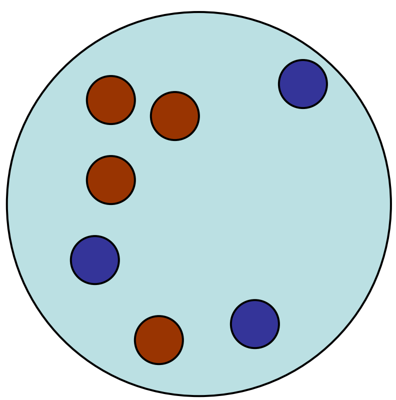
The fundamentals



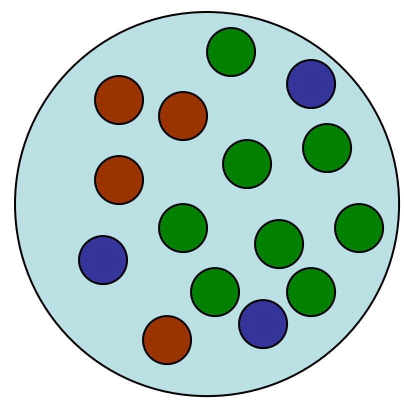
1. Population of individuals (solutions)



2. Evaluation of fitness



3. Selection



4. Reproduction

Repeat…

History

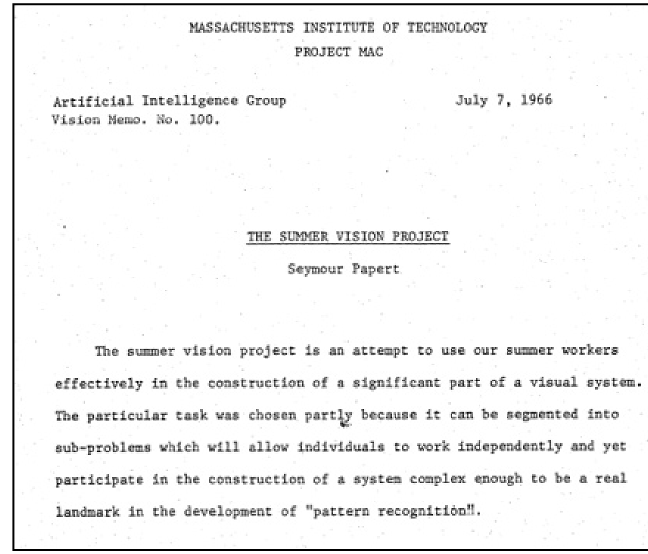
A branch of A-Life

Arises from Artificial Intelligence

Strong AI: we can model intelligence

Weak AI: we can produce the effect of intelligence

A-Life: perhaps we can't even engineer intelligence



MASSACHUSETTS INSTITUTE OF TECHNOLOGY
PROJECT MAC

Artificial Intelligence Group July 7, 1966
Vision Memo. No. 100.

THE SUMMER VISION PROJECT

Seymour Papert

The summer vision project is an attempt to use our summer workers effectively in the construction of a significant part of a visual system. The particular task was chosen partly because it can be segmented into sub-problems which will allow individuals to work independently and yet participate in the construction of a system complex enough to be a real landmark in the development of "pattern recognition".

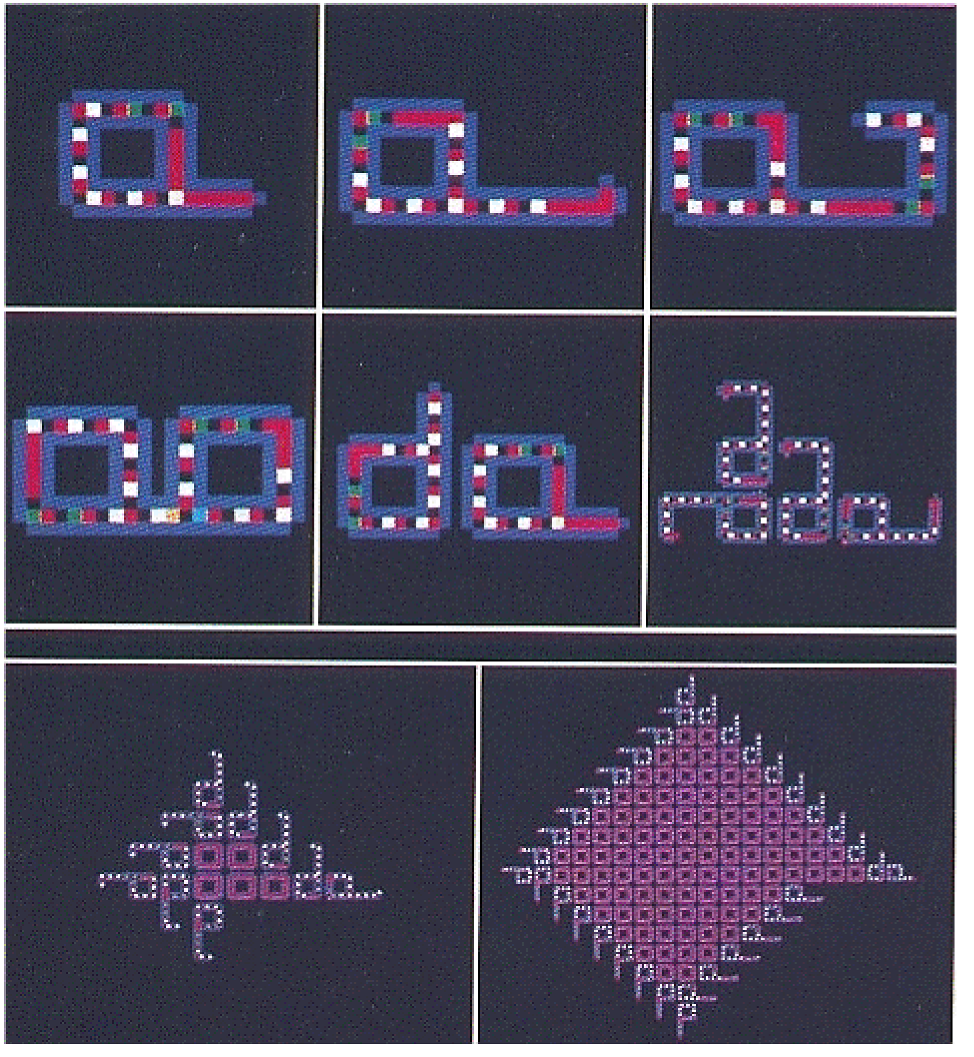
Project MAC

Early experiments

Following von Neumann…

Chris Langton's self-replicating cellular automata (CA) loops

Fundamentally, not evolutionary computation

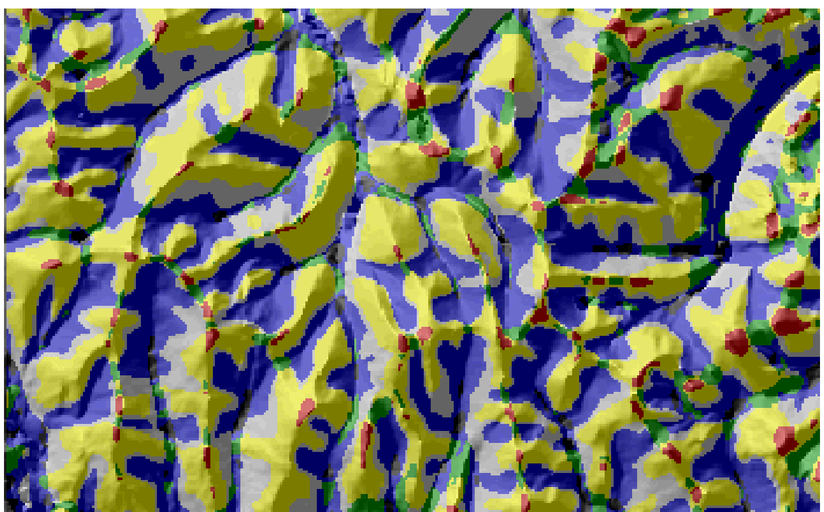


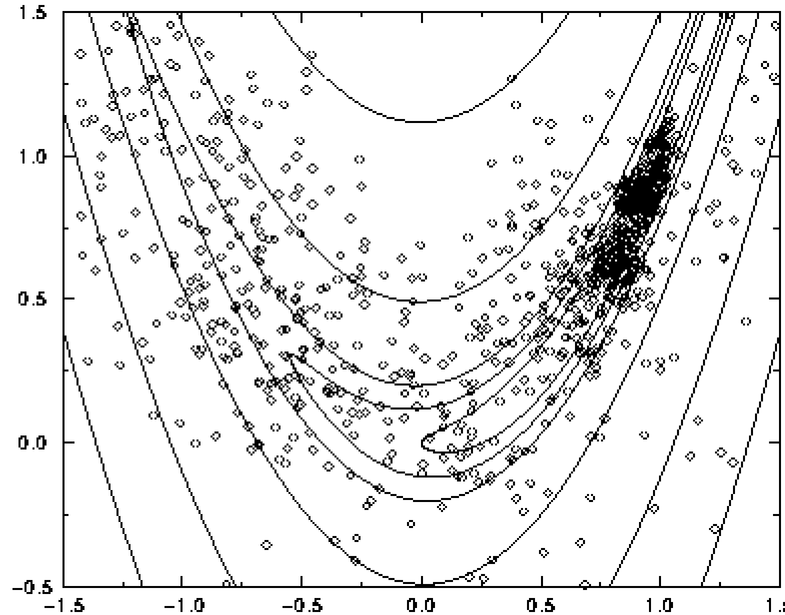
Stochastic optimization

John Holland (develops GA 1962-75)
hybrid of A-life and classical optimization

Simulated annealing
- a gradual reduction in random energy
- samples a large space
- attempts to find global maxima/minima

GA is a very similar technique (search strategy)





Solution-space sampling of a simulated-annealing run

The inadequacy of hill-climbing for global maxima

GA Specifics

Population

- randomly initialized
- genome often a bit-string,
- anything that can be combined

Decoding

- genotype to phenotype

Evaluation

- fitness (objective) function
- single-valued!
- goodness as a number?

Selection

- proportional
- tournament
- elitism

Breeding

- multiple genomes combined
- (can be more than two)
- mutation and crossover

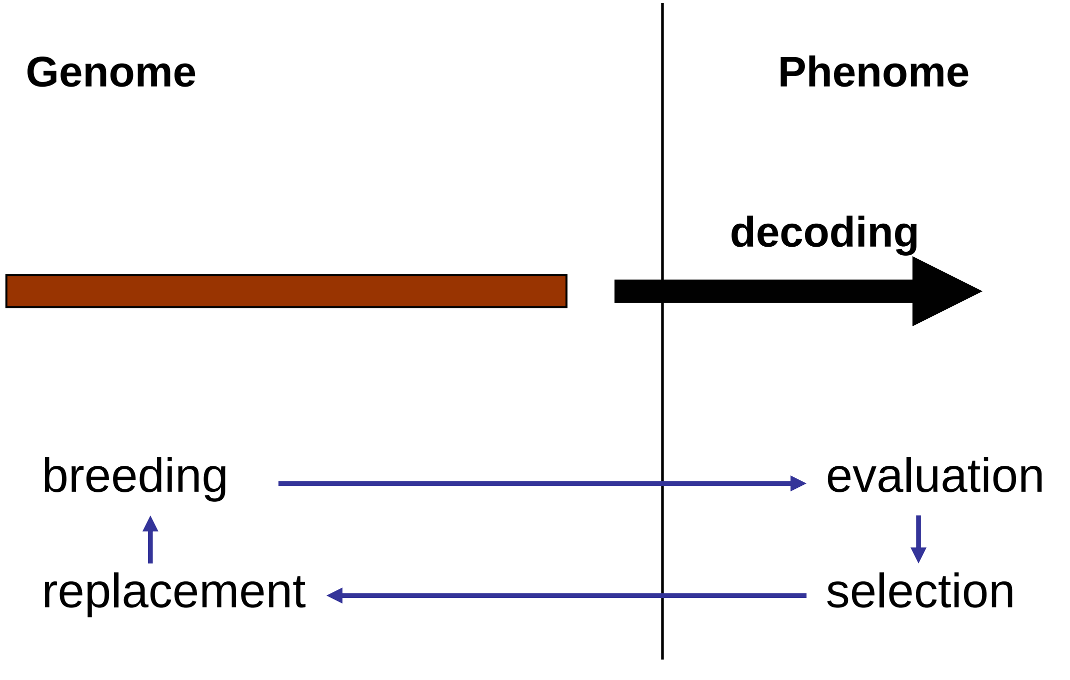
Replacement

- replace entire population
- or just the worst
- must retain diversity

Genotype vs. Phenotype

Problem specified phenotype

Encoding into genome is up to you
Must be extremely clever (most are bad)



Crossover and Mutation

Parents' genomes are combined

Some methods favor one over the other

Mutation is relatively rare

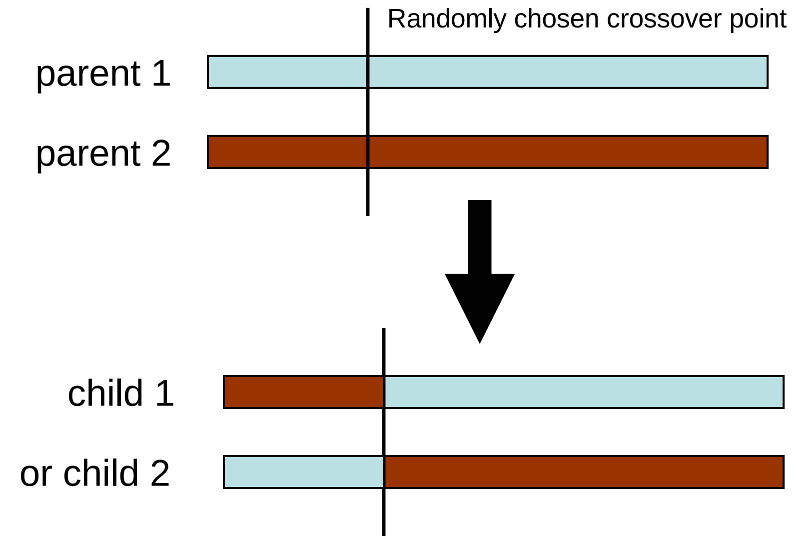
The importance of genetic diversity

Population size

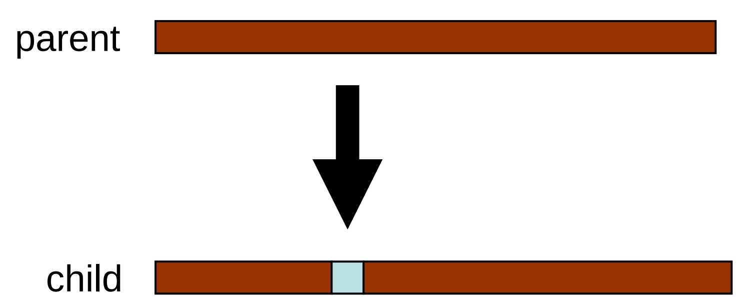
Preserving "strategy"

Why not average?

Crossover



Mutation



Why operate on genomes?

Why not do combinatorial population work directly on phenotypes?

conduciveness to operations

generality of operations

nonlinearity of decoding

power of decoding

correlation of genes vs. traits

Genesys: The John Muir Trail

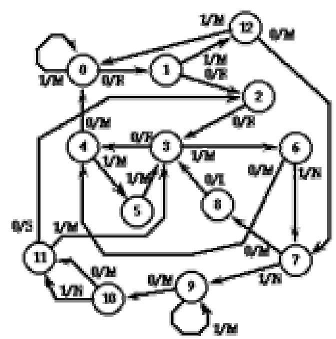
David Jefferson, 1992

evolving ant brains to follow a trail of food
fitness is amount eaten in a certain time

in any state left, right, forward, no movement
and exit to another state
number of states is crucial

behavior expressed as Finite State Machine (FSA)

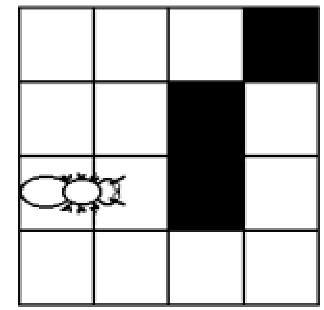
Old State	Input	New State	Action
00	0	00	01
00	1	01	11
01	0	01	01
01	1	11	11
10	0	10	10
10	1	11	11
11	0	10	10
11	1	10	11

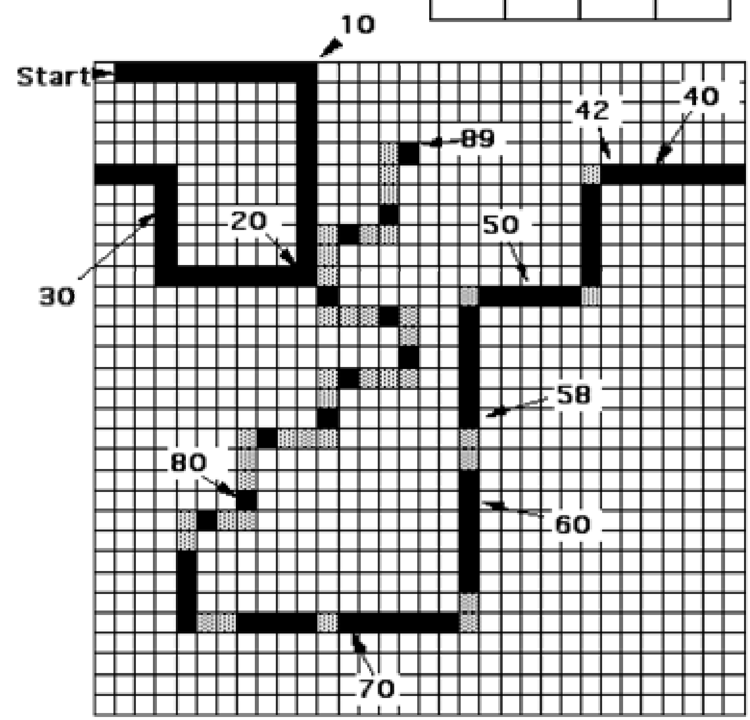


Genome

00 0001 0111 0101 1111 1010 1111 1010 1011

body in some color,
head in another





Genetic Programming

John Koza c. 1992

Represents genomes as “programs”
(trees of operations and values)

Can even define and call subroutines

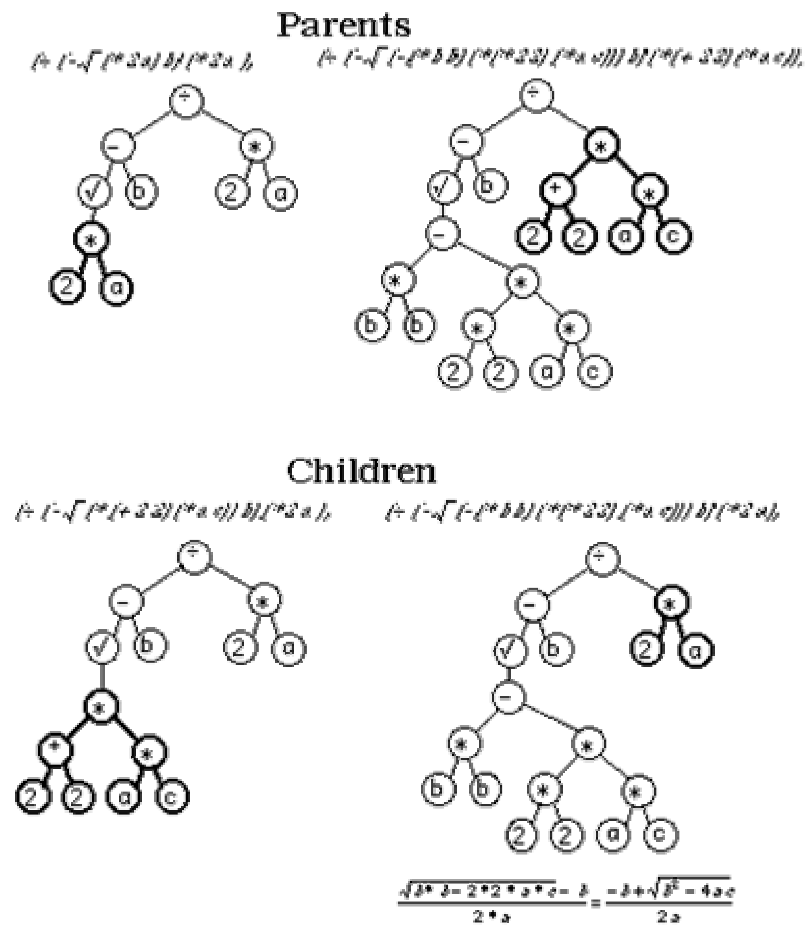
Phenome is the production of running the program

Crossover is grafting of branches from one genome to another

Not usually any mutation

Steady-state population (no specific generations)

Space of exploration very broad



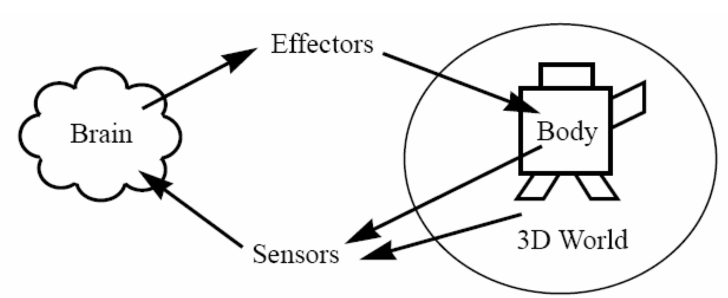
Compelling work: Karl Sims

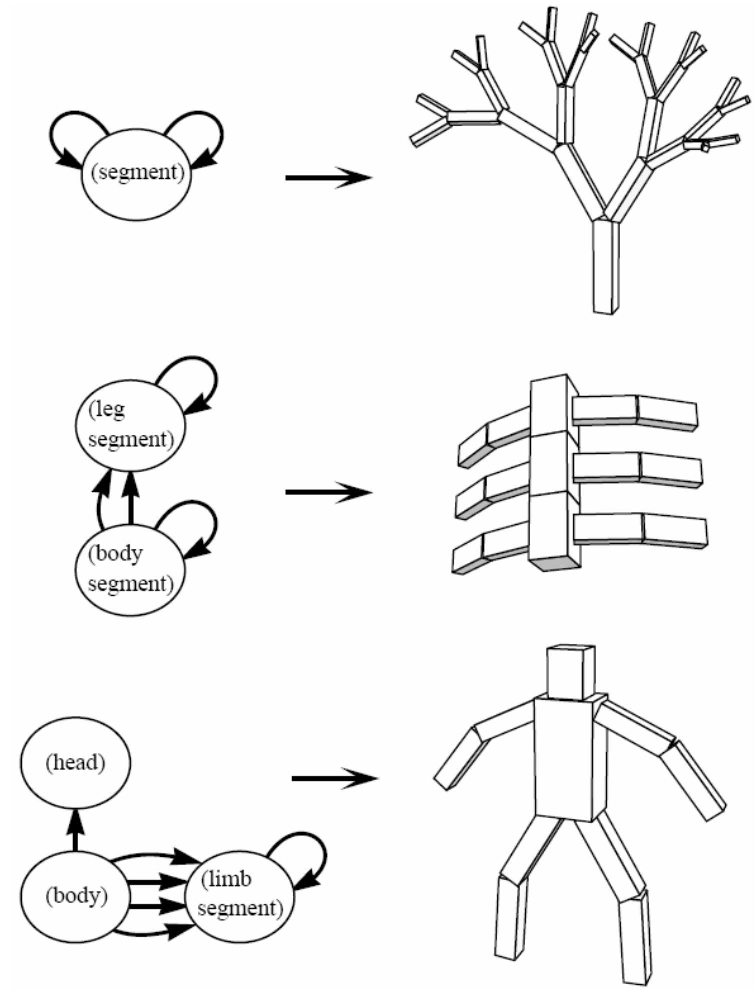
Evolved virtual creatures, 1994
(like all great work, looks easy)

genome coded for both form and behavior
sensors and effectors (muscles)
and a neural “brain”

incredibly complicated genome
multiple nested directed graphs

careful physics simulation as environment



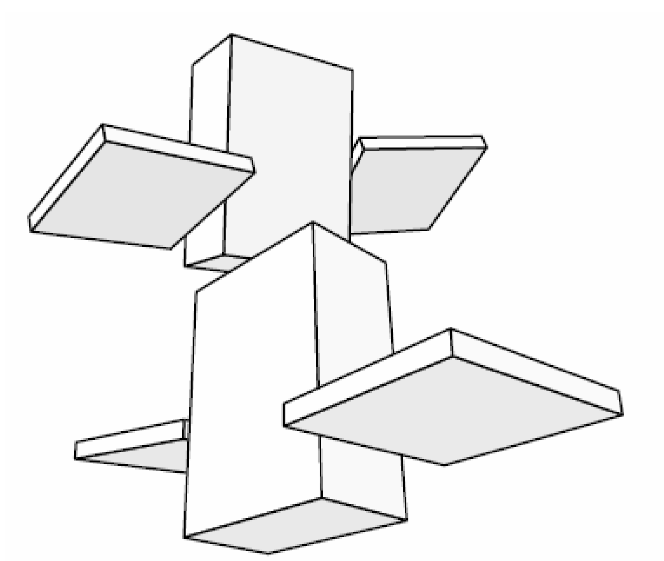


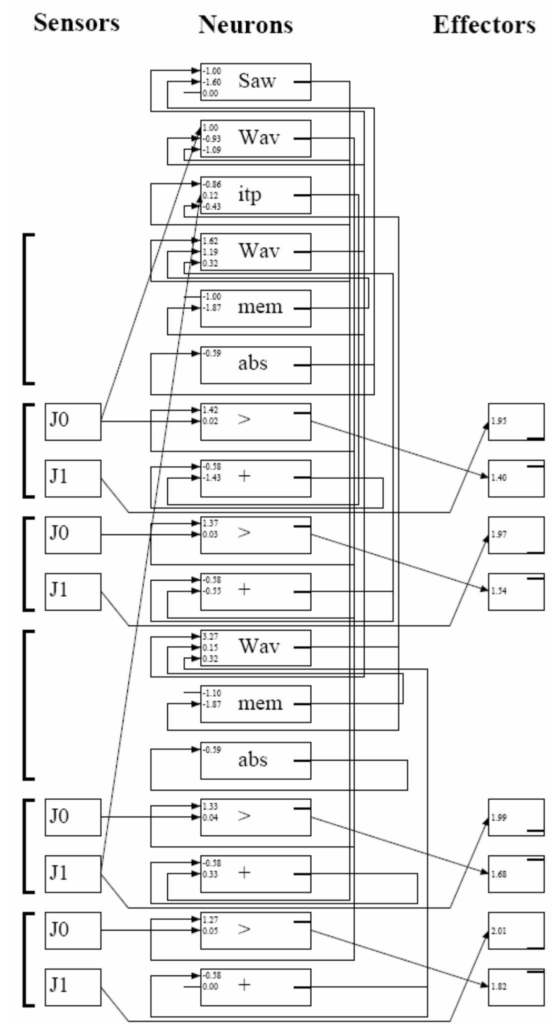
Compelling work: Karl Sims

ran on a massively parallel machine

selected for results of behaviors

also direct competition



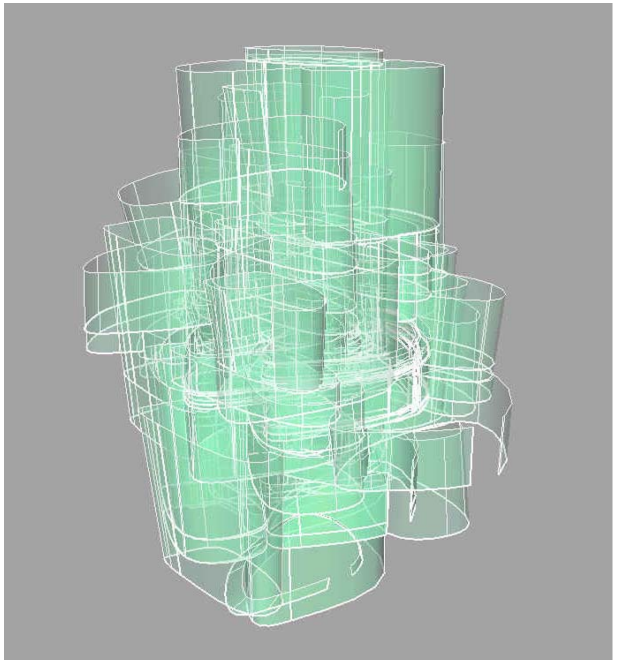


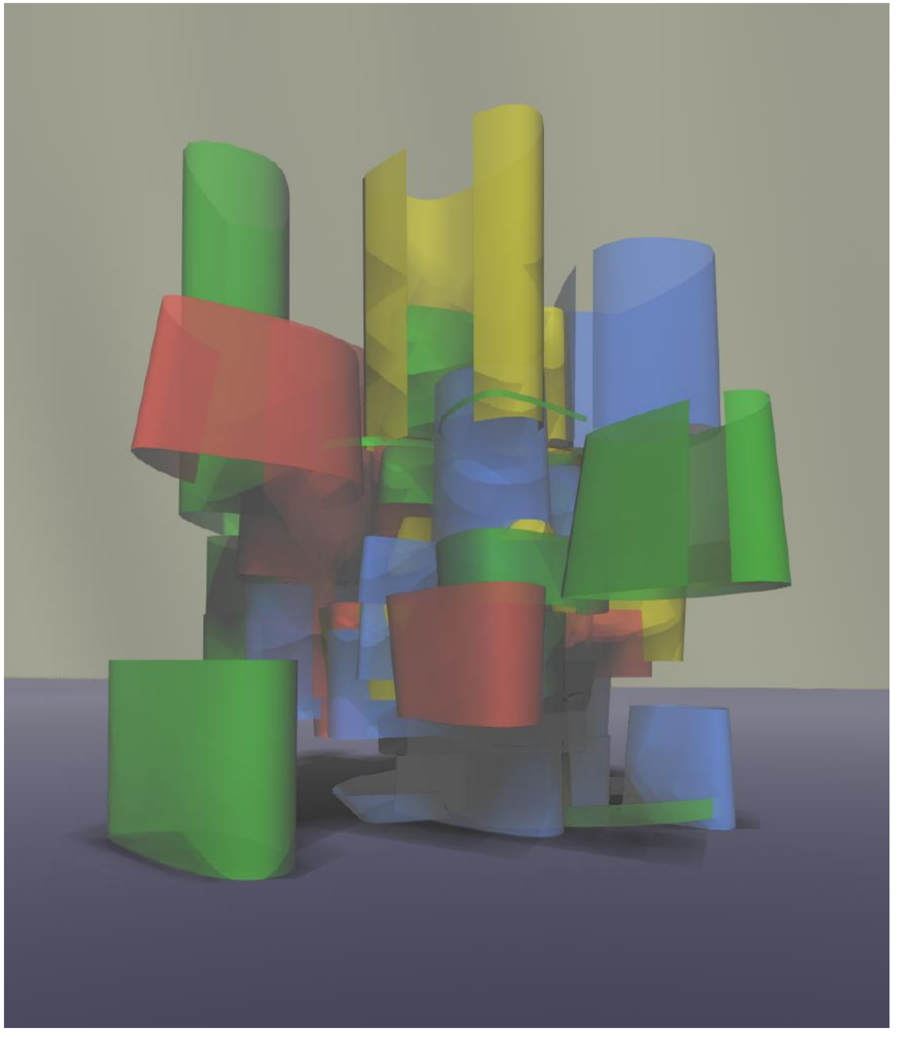
AgencyGP

Genetic programming for growing extruded NURBS clusters

Testa, O'Reilly, and me

Plugin to Maya (3.x at the time)



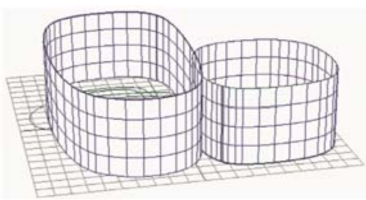
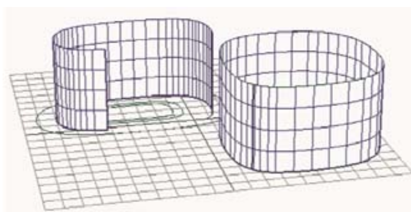
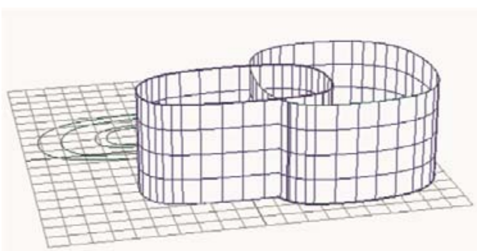
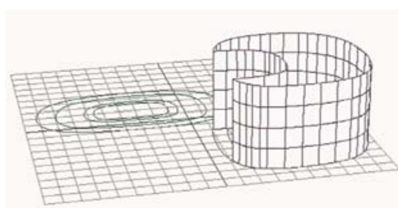


AgencyGP

Individuals

A “program” of transformations and boolean operations on NURBS curves

Colors for program or other division of control

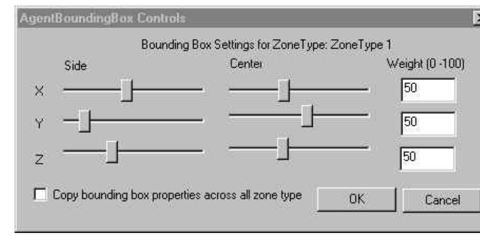
Rotate, translate, scale, extrude, union, intersect, difference.

Agents

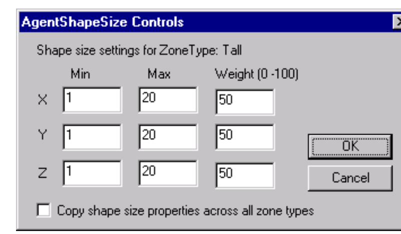
Fitness a weighted function of the evaluation of arbitrarily many “agents”

A framework…

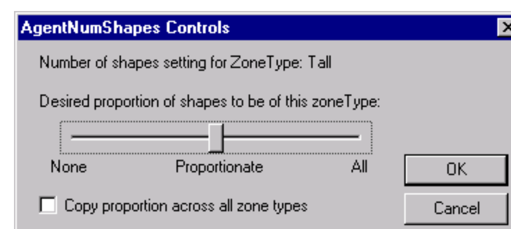
Bounding box



Size by type



Number of shapes



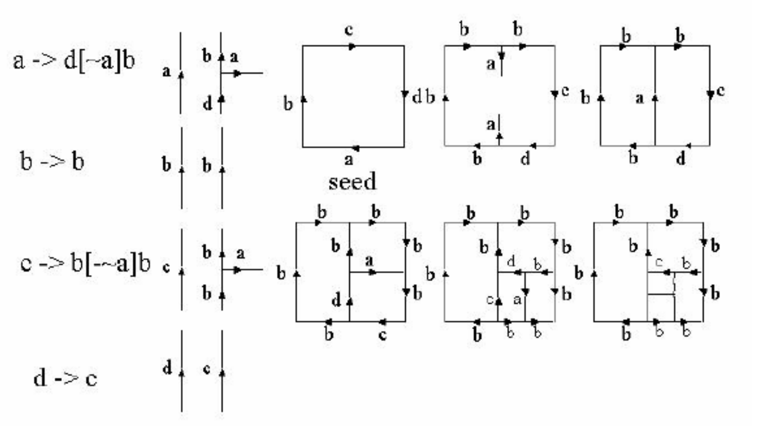
GenR8

O'Reilly & Hemberg

Evolving grammars for surfaces

A well-known decoding: L-Systems

User-defined fitness, attractors, repellers



A shared environment: Tierra

Tom Ray's 1991 TIERRA *Synthetic Life program*

Organisms as strings of instructions

Executed directly in a virtual machine

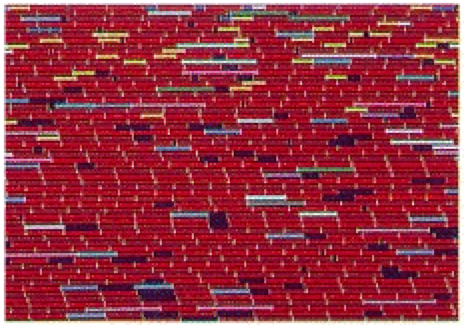
To reproduce, had to reproduce themselves in memory

Organisms became the environment

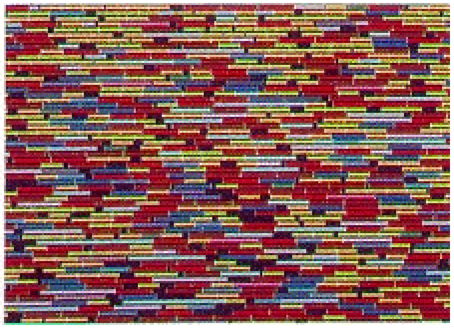
Competition was innate to the system

No external selection mechanism required

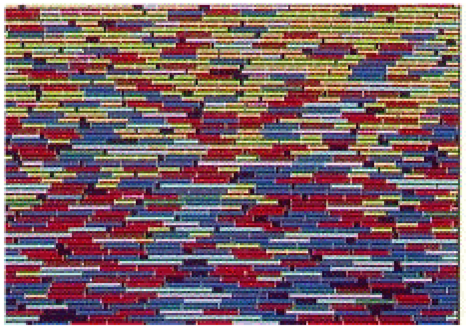
Strategies developed (hosts, parasites - biodiversity)



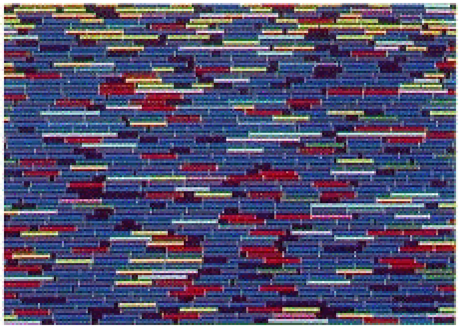
(1) Hosts, red, are very common. Parasites, yellow, have appeared but are still rare.



(2) Parasites have become very common. Hosts immune to parasites, blue, have appeared.



(3) Immune hosts now dominate memory, while parasites and susceptible hosts decline.



(4) The parasites will soon be driven to extinction.

Hard problems

What makes a problem hard?

Magnitude and shape of solution space
(discontinuities, any pathology imaginable)

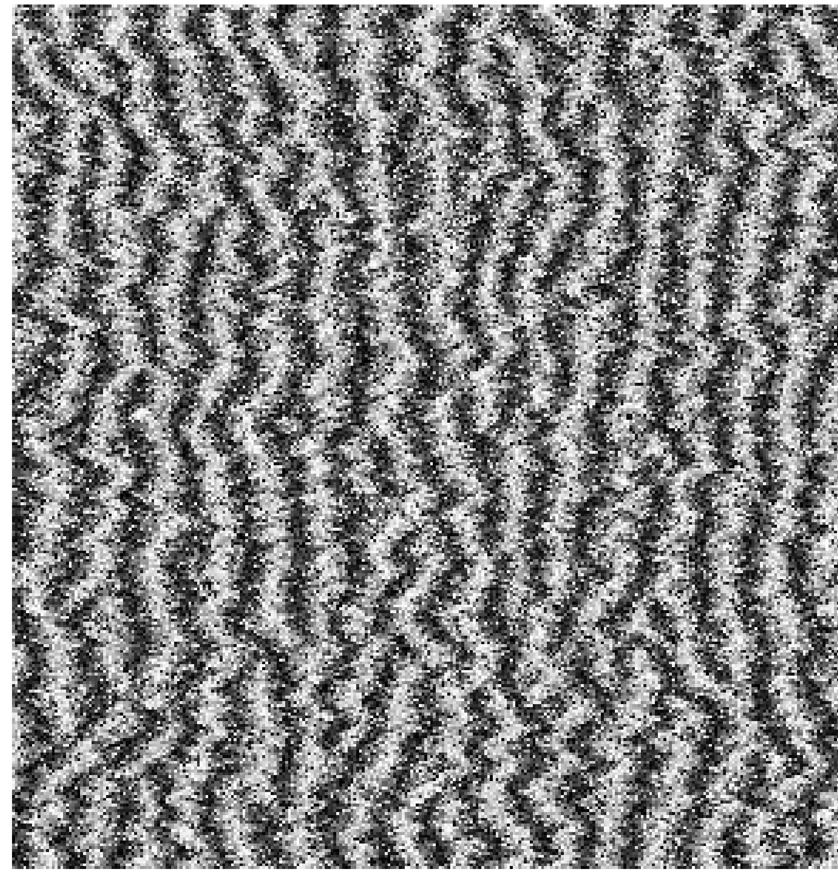
Genomic representation that cover the space exactly

Genomic surface area

The difficulty of fitness functions
(weighted concerns, creativity?)

Where are we today?

There is no general solver



GAs vs. interactive-selection systems

Evolutionary computation for design

The limited role of optimization

The fitness problem – what are we selecting for?
(and if we know it, why don't we just design it?)

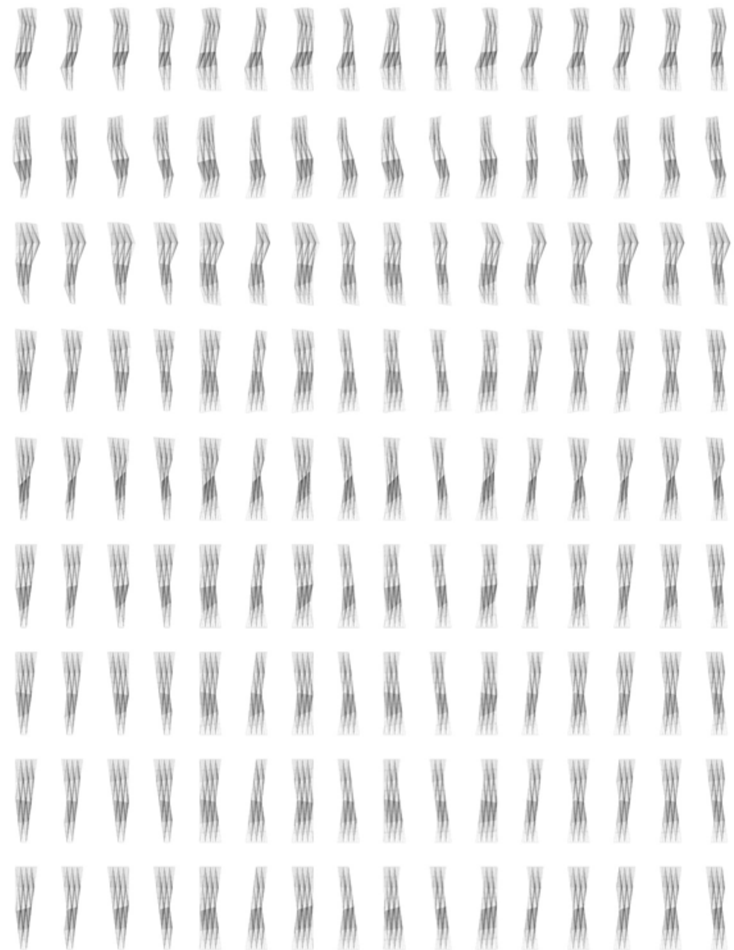
Aesthetic criteria (exploration vs. optimization)

A different relationship to solution space

The parameters frame the solution

Direct optimization is better

Why do these capture the imagination?



Results from GenR8