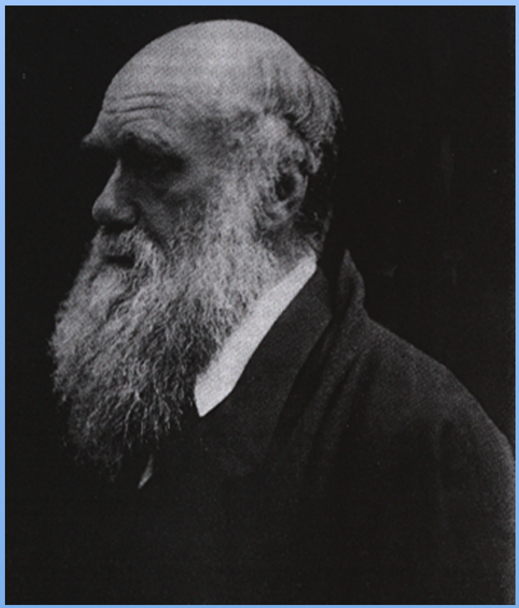
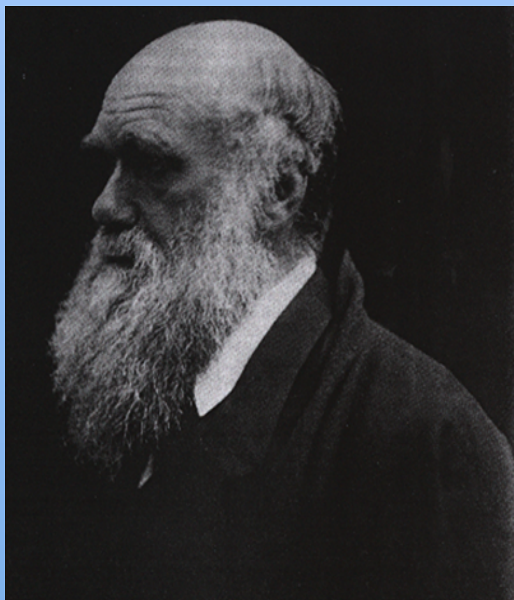
Evolution by Natural Selection



Charles Darwin, 1809−1882

Evolution by Natural Selection

in computers



Charles Darwin, 1809–1882

John Holland

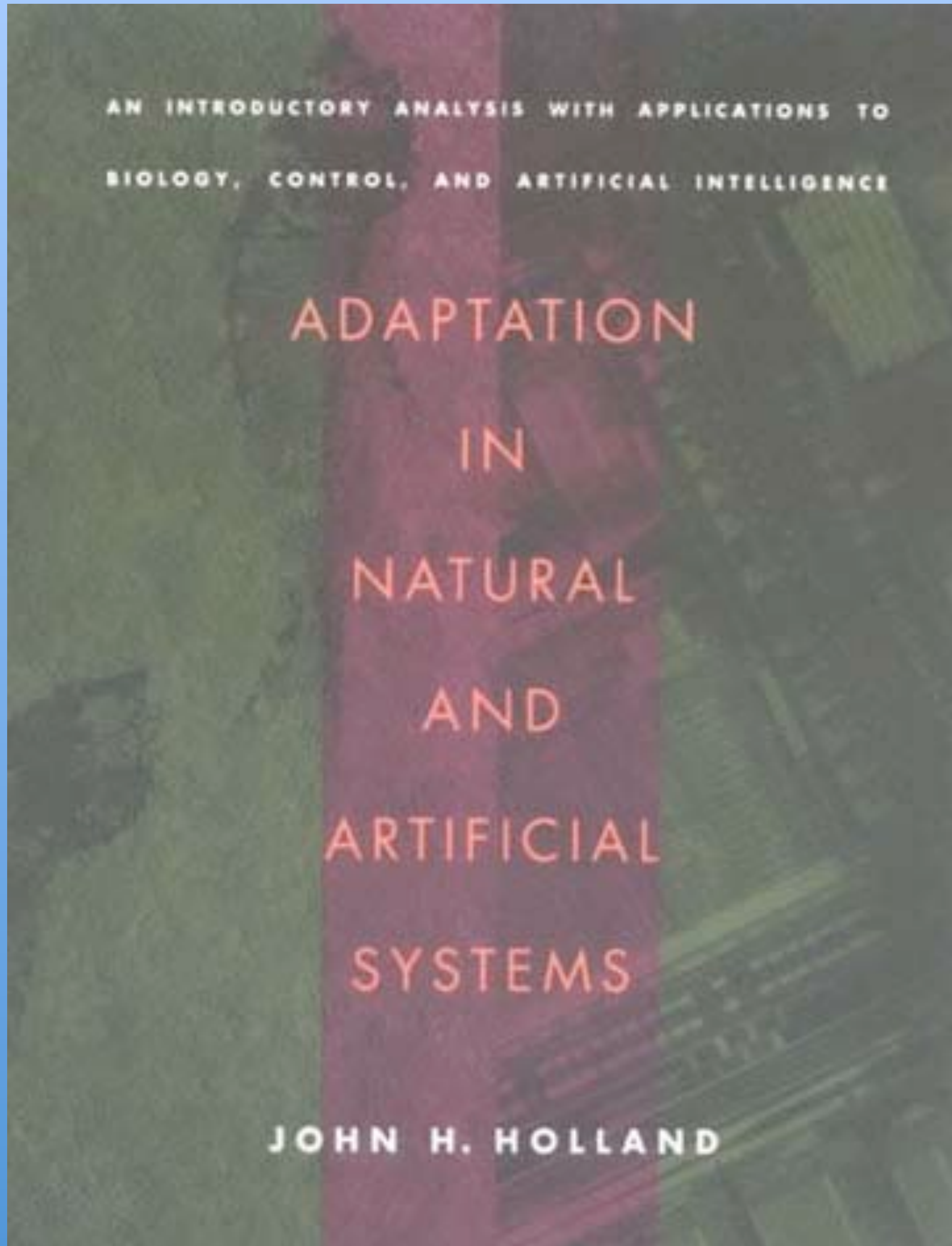
- Organisms inherit traits from parents

Computer (e.g., programs)

- Traits are inherited with some variation, via mutation and sexual recombination

- Due to competition for limited resources, the organisms best adapted to the environment tend to produce the most offspring.

- This way traits producing adapted individuals spread in the population


AN INTRODUCTORY ANALYSIS WITH APPLICATIONS TO
BIOLOGY, CONTROL, AND ARTIFICIAL INTELLIGENCE
ADAPTATION IN NATURAL AND ARTIFICIAL SYSTEMS
JOHN H. HOLLAND

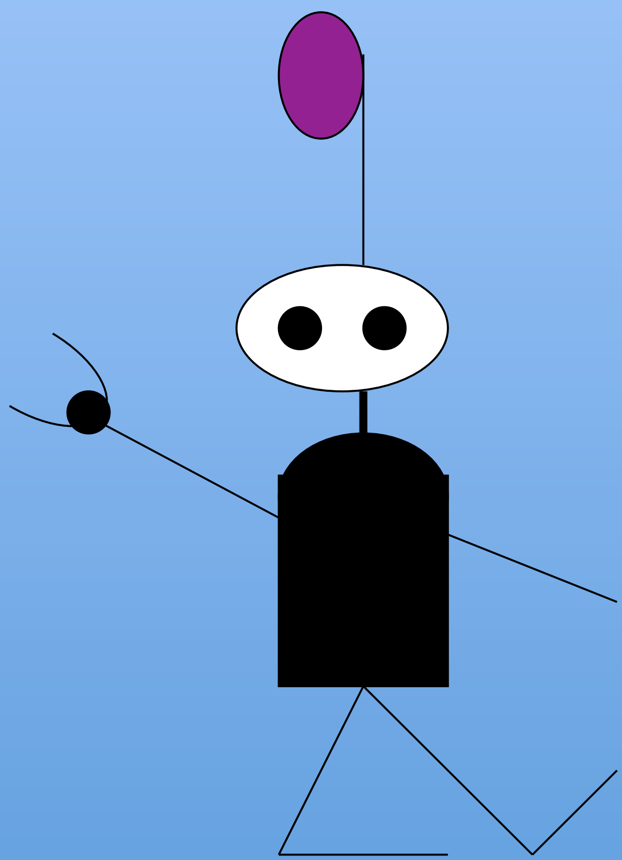
Some real-world uses of genetic algorithms

- **Designing parts of aircraft (GE and Boeing)**
- **Spacecraft antenna design (NASA)**
- **Assembly line scheduling (John Deere Co.)**
- **Automated drug discovery (several companies)**
- **Fraud detection (credit cards, financial trading)**
- **Automated analysis of satellite images (Los Alamos National Lab)**
- **Generation of realistic computer animation (*Lord of the Rings: The Return of the King* and *Troy*)**

Genetic Algorithm Example:

Evolving a Control Program for a Virtual "Robot"

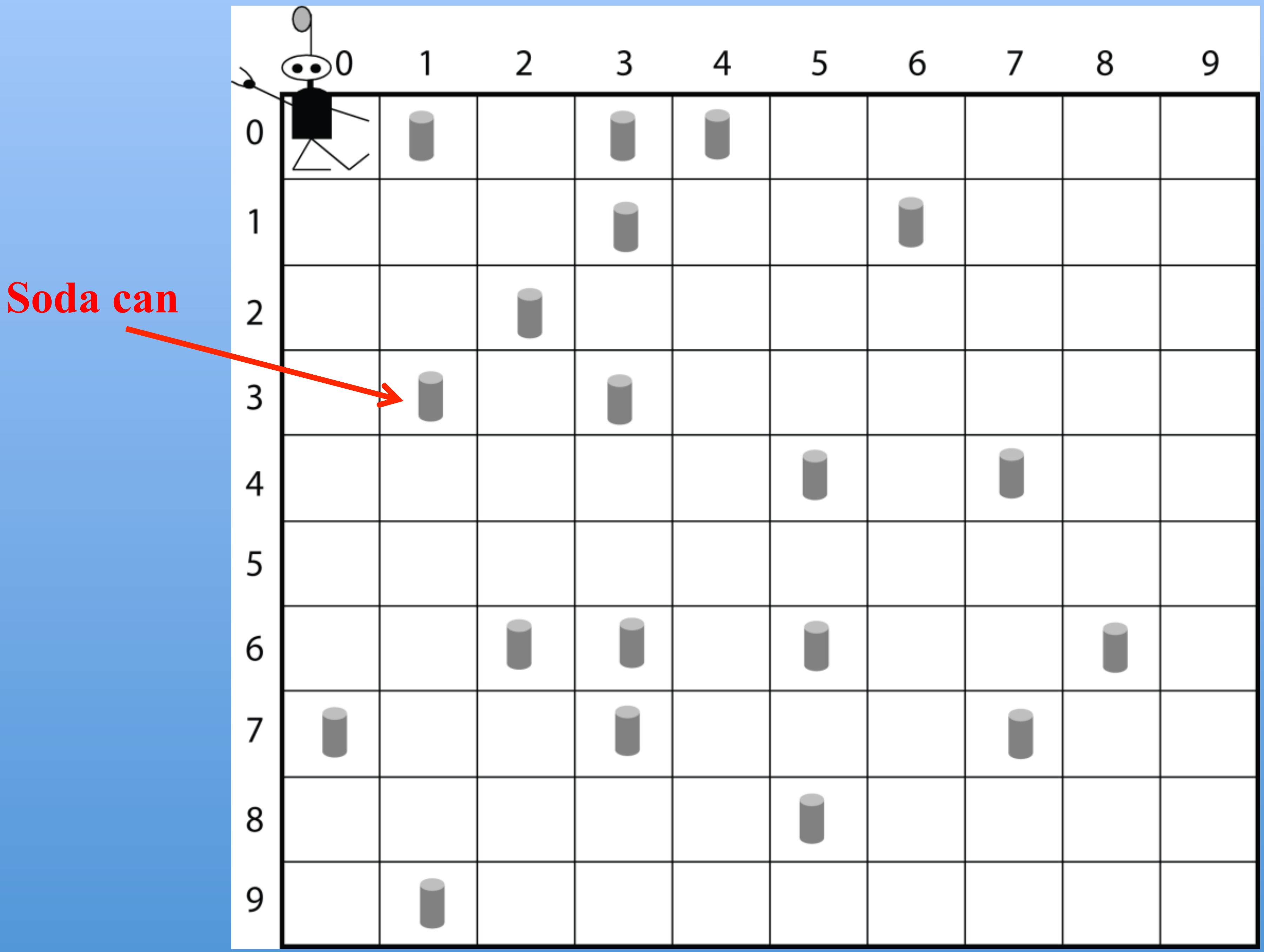
Robby:
The Virtual Soda Can Collecting Robot
(Mitchell, 2009)



Herbert:
The Soda Can Collecting Robot
(Connell, Brooks, Ning, 1988)

http://cyberneticzoo.com/?p=5516

Robby's World

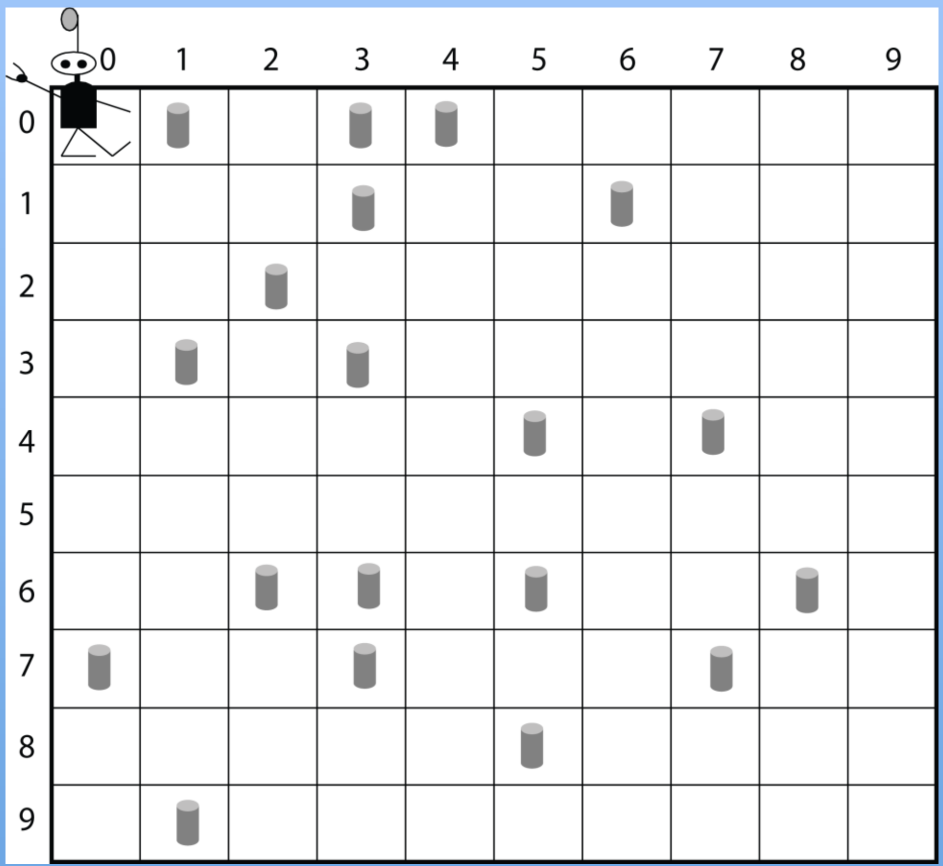


What Robby Can See and Do

Input:
Contents of **N**orth, **S**outh, **E**ast,
West, **C**urrent

Possible actions:

```
Move N

Move S

Move E

Move W

Move random

Stay put

Try to pick up can
```



Rewards/Penalties (points):

Picks up can: **10**

Tries to pick up can on empty site: **-1**

Crashes into wall: **-5**

Robby's Score: Sum of rewards/penalties

Goal: Use a genetic algorithm to evolve a control program (i.e., *strategy*) for Robby.

What is a “strategy”?

Strategy: A set of rules that specifies an ***action*** for every possible ***situation***.

Possible Situations = possible inputs to Robby

North	***South***	***East***	***West***	***Current Site***	***Action***
Empty	Empty	Empty	Empty	Empty	
Empty	Empty	Empty	Empty	Can	

.

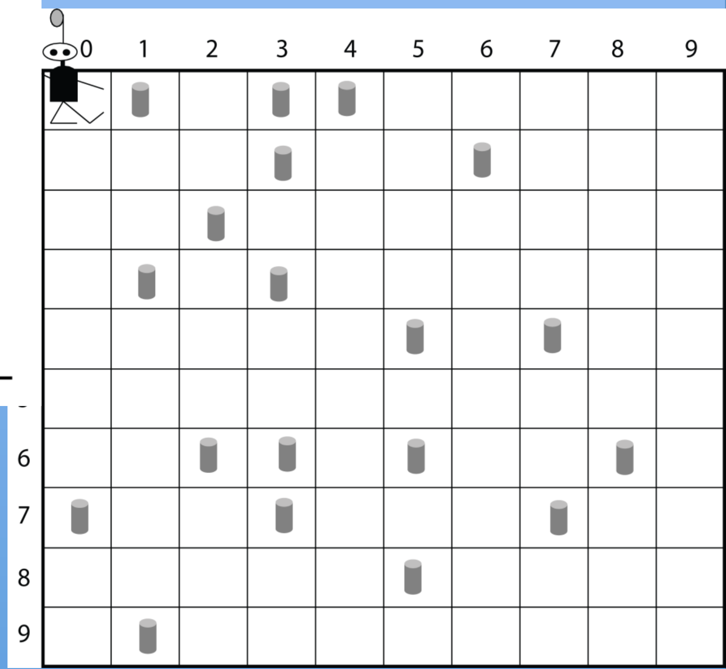
.

.

North: 3 possibilities (Empty, Can, Wall) × South : 3 possibilities × East: 3 possibilities × West: 3 possibilities × Current Site : 3 possibilities $= 3 \times 3 \times 3 \times 3 \times 3 = 243$

One Example Strategy

	Situation					Action
	North	*South*	*East*	*West*	*Current Site*	
1	Empty	Empty	Empty	Empty	Empty	MoveNorth
2	Empty	Empty	Empty	Empty	Can	MoveEast
3	Empty	Empty	Empty	Empty	Wall	MoveRandom
4	Empty	Empty	Empty	Can	Empty	PickUpCan
.	⋮	⋮	⋮	⋮	⋮	⋮
.	Wall	Empty	Can	Wall	Empty	MoveWest
.	⋮	⋮	⋮	⋮	⋮	⋮
243	Wall	Wall	Wall	Wall	Wall	StayPut



Question: What will Robby’s score be after following this strategy for three time steps?

Answer: −15

Encoding a Strategy

	Situation					Action
	North	South	East	West	Current Site	
1	Empty	Empty	Empty	Empty	Empty	MoveNorth
2	Empty	Empty	Empty	Empty	Can	MoveEast
3	Empty	Empty	Empty	Empty	Wall	MoveRandom
4	Empty	Empty	Empty	Can	Empty	PickUpCan
.	⋮	⋮	⋮	⋮	⋮	⋮
.	Wall	Empty	Can	Wall	Empty	MoveWest
.	⋮	⋮	⋮	⋮	⋮	⋮
243	Wall	Wall	Wall	Wall	Wall	StayPut

Encoding a Strategy

	Action
1	MoveNorth
2	MoveEast
3	MoveRandom
4	PickUpCan
.	⋮
.	MoveWest
.	⋮
243	StayPut

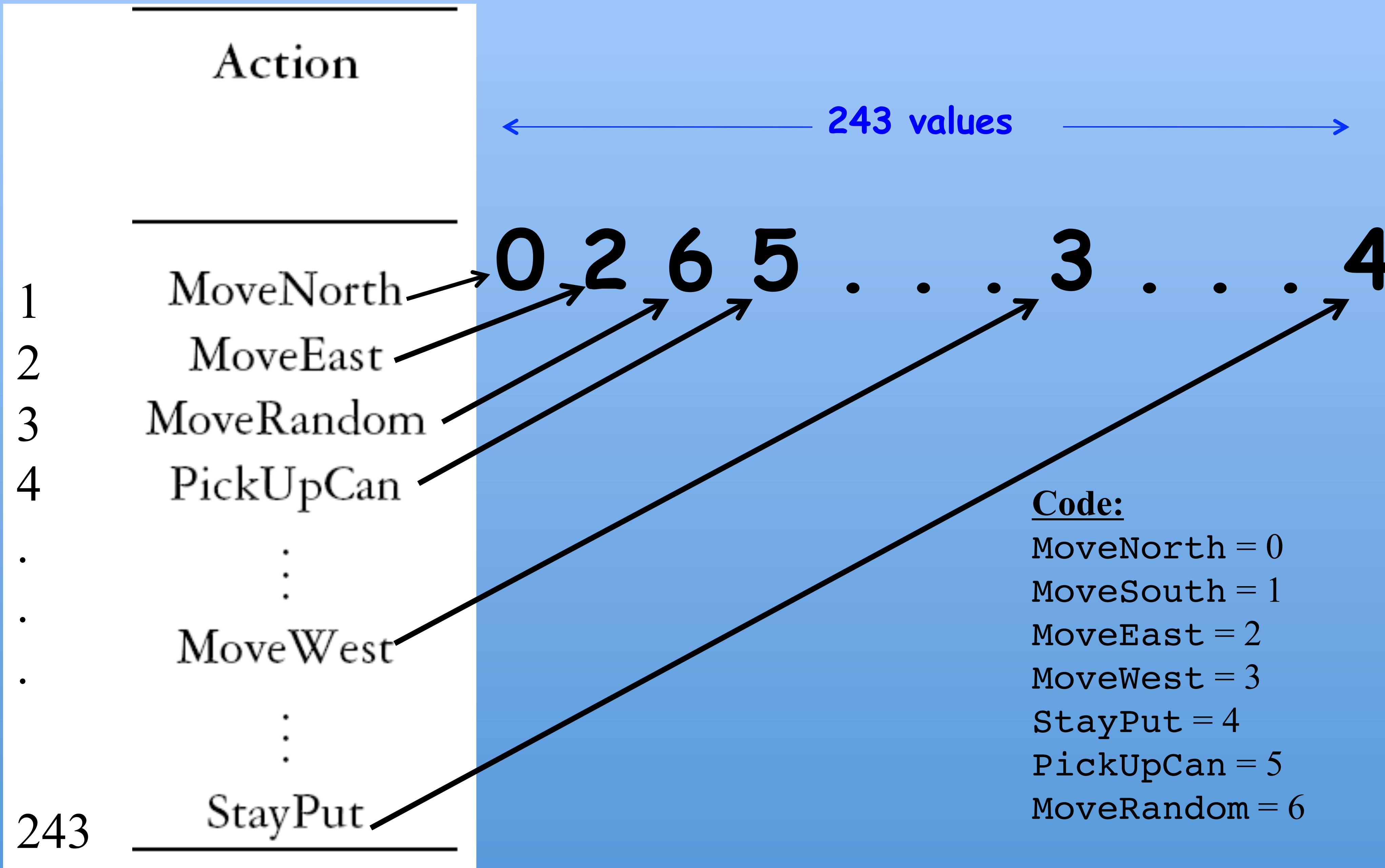
Encoding a Strategy

	Action
1	MoveNorth
2	MoveEast
3	MoveRandom
4	PickUpCan
.	⋮
.	MoveWest
.	⋮
243	StayPut

Code:

`MoveNorth` = 0
`MoveSouth` = 1
`MoveEast` = 2
`MoveWest` = 3
`StayPut` = 4
`PickUpCan` = 5
`MoveRandom` = 6

Encoding a Strategy



Question: How many possible strategies are there in our representation?

243 values

0 2 6 5 . . . 3 . . . 4

7 possible actions for each position:

$7 \times 7 \times 7 \times \ldots \times 7$

Answer: 7^{243}

Goal: Have GA search intelligently in this vast space for a good strategy

Genetic algorithm for evolving strategies

1. Generate 200 random strategies (i.e., programs for controlling Robby)

2. For each strategy, calculate fitness (average reward minus penalties earned on random environments)

3. The strategies pair up and create offspring via “sexual recombination” with random mutations — the fitter the parents, the more offspring they create.

4. Keep going back to step 2 until a good-enough strategy is found!

Random Initial Population

Individual 1:
23300323421630343530546006102562515114162260435654334066511514
15650220640642051006643216161521652022364433363346013326503000
40622050243165006111305146664232401245633345524126143441361020
150630642551654043264463156164510543665346310551646005164

Individual 2:
16411343121025360340361241431201104235462525304202044516433665
61035322153105131440622120614631432154610256523644422025340345
30502005620634026331002453416430151631210012214400664012665246
351650154123113132453304433212634555005314213064423311000

Individual 3:
20423344402411226132136452632464212206122122252660626144436125
32512664061335340153411110206164226653145522540234051155031302
22020065445125062206631426135532010000400031640130154160162006
134440626160505641421553133236021503355131253632642630551

.

.

.

Individual 200:
34632525136001012225612106043301135205155320130656005322235043
32425064124255265534635345523053326612010632124554423440613654
30246240160663016464641103026540006334126150352262106063624260
550616616344255124354464110023463330440102533212142402251

Parent 1:

16411343121025360340361241431201104235462525304202044516433665
61035322153105131440622120614631432154610256523644422025340345
30502005620634026331002453416430151631210012214400664012665246
3516501541231131324533044332126345550053142130644233 11000

Parent 2:

20423344402411226132136452632464212206122122252660626144436125
32512664061335340153411110206164226653145522540234051155031302
22020065445125062206631426135532010000400031640130154160162006
134440626160505641421553133236021503355131253632642630551

Parent 1:

16411343121025360340361241431201104235462525304202044516433665
61035322153105131440622120614631432154610256523644422025340345
30502005620634026331002453416430151631210012214400664012665246
3516501541231131324533044332126345550053142130644233110000

Parent 2:

20423344402411226132136452632464212206122122252660626144436125
32512664061335340153411110206164226653145522540234051155031302
22020065445125062206631426135532010000400031640130154160162006
1344406261605056414215531332360215033551312536326426305510

Child:

5

1641134312102536034036124143120110423546252530420204445 6433665
61035322153105131440622120614631432154610256523644422025340345
30502005620634026331002456135532010000400031640130154160162006
1344406261605056414215531332360 503355131253632642630551

1

Maximum possible fitness ≈ 500

Maximum possible fitness $\approx$ 500

- There are 100 squares total, and each environment starts out with about 50 cans.

- Each can is worth 10 points

My hand-designed strategy:

"If there is a can in the current site, pick it up."

"Otherwise, if there is a can in one of the adjacent sites, move to that site."

"Otherwise, choose a random direction to move in."

Average fitness of this strategy: **346**
(out of max possible ≈ 500)

Average fitness of GA evolved strategy:
486
(out of max possible ≈ 500)

My hand-designed strategy:

"If there is a can in the current site, pick it up."

"Otherwise, if there is a can in one of the adjacent sites, move to that site."

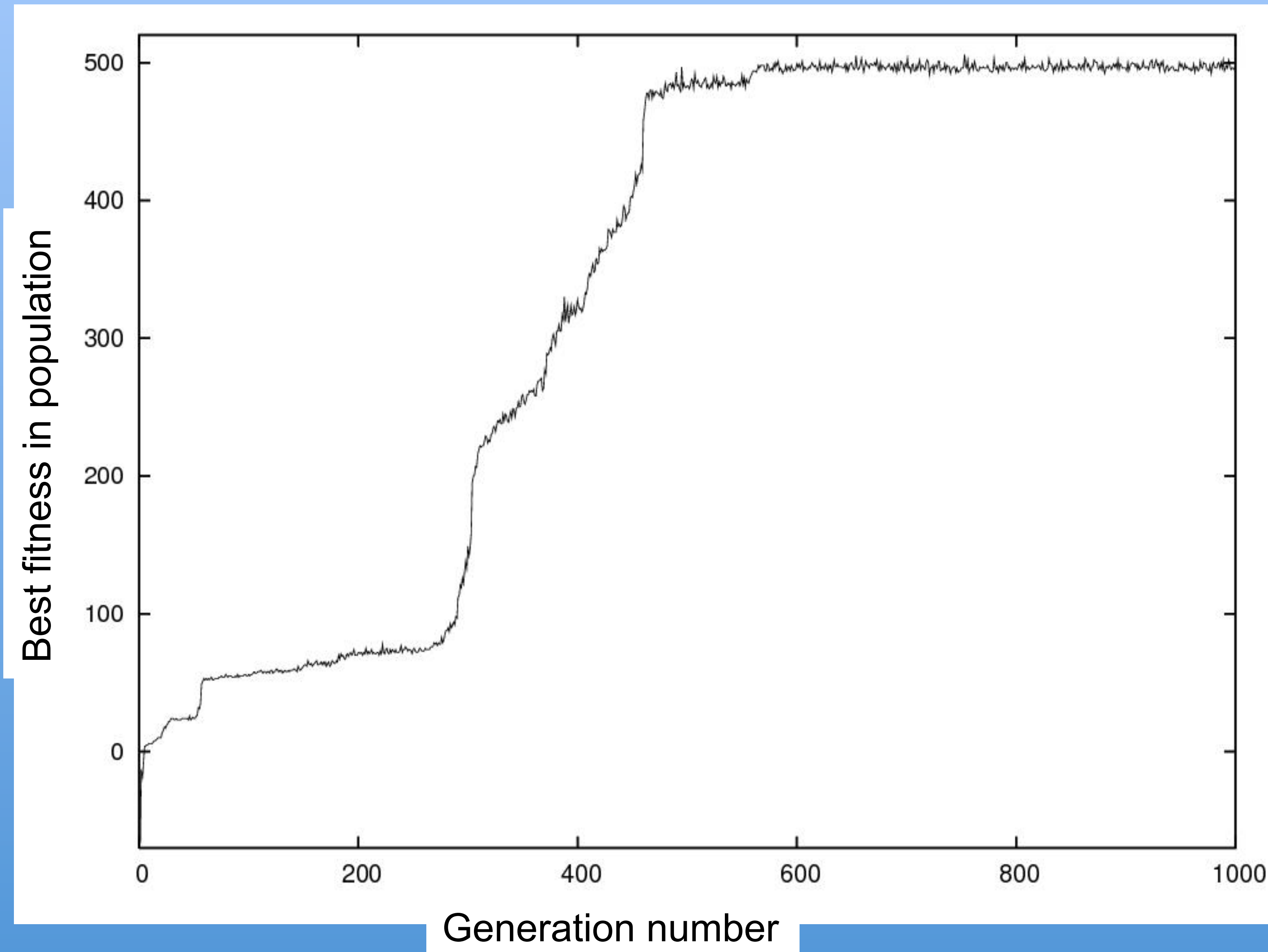
"Otherwise, choose a random direction to move in."

Average fitness of this strategy: **346**
(out of max possible ≈ 500)

???

Average fitness of GA evolved strategy:
486
(out of max possible ≈ 500)

One Run of the Genetic Algorithm (C version)



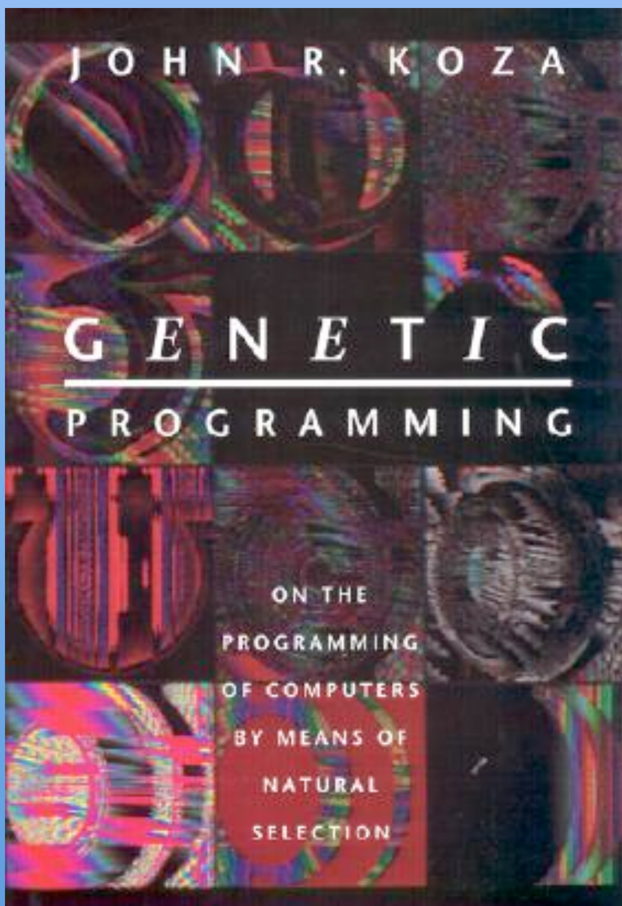
Principles of Evolution Seen in Genetic Algorithms

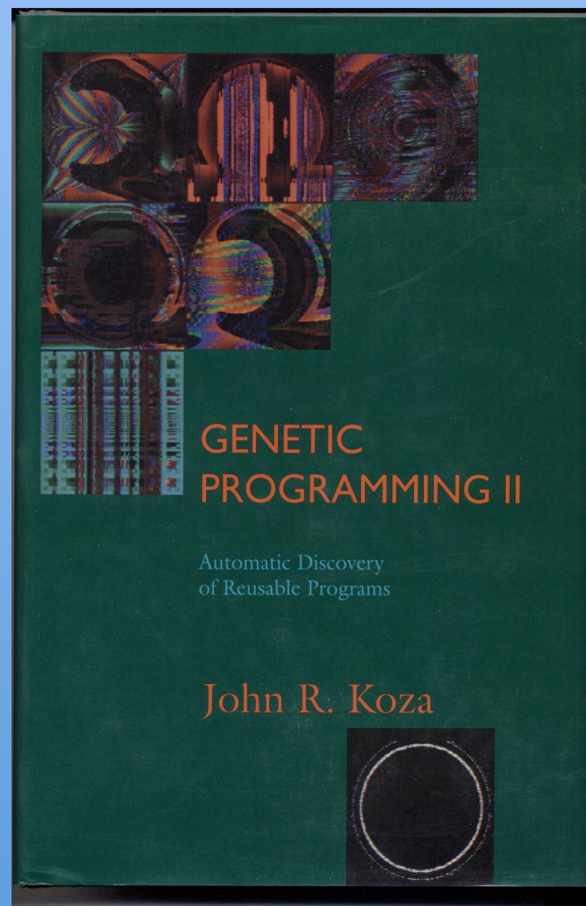
- Natural selection works!
- Evolution proceeds via periods of stasis “punctuated” by periods of rapid innovation
- *Exaptation* is common. **Exaptation**: “shifts in the function of a trait during evolution.
- Dynamics and results of evolution are unpredictable and hard to analyze

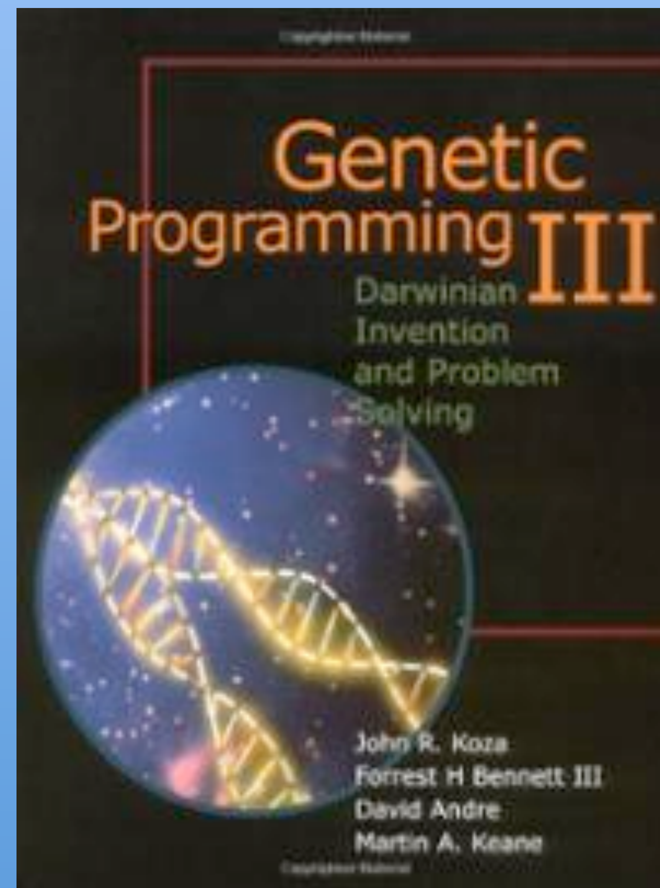
Genetic Programming (John Koza, 1990s)

John Koza

Genetic Programming
(John Koza, 1990s)

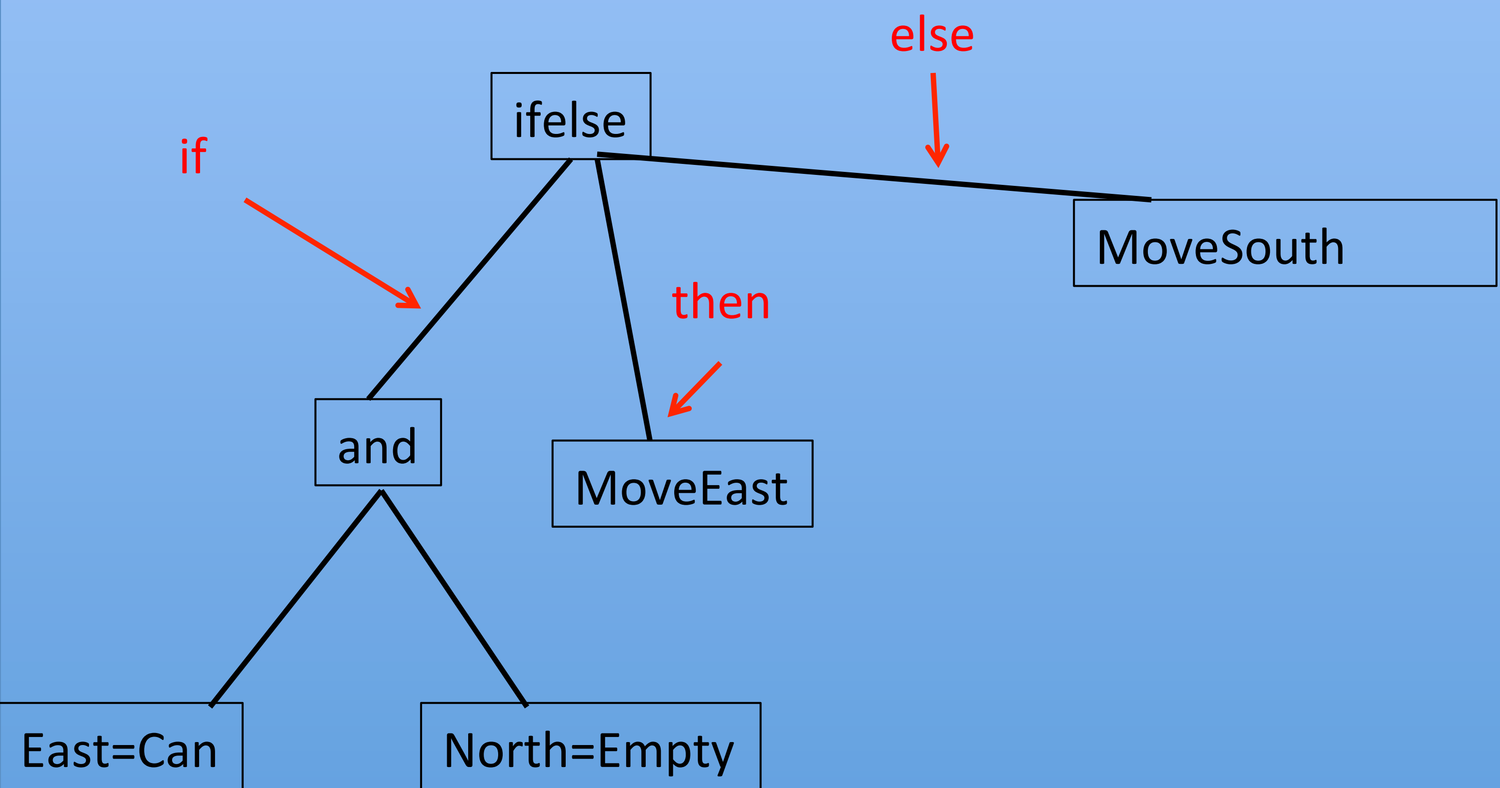






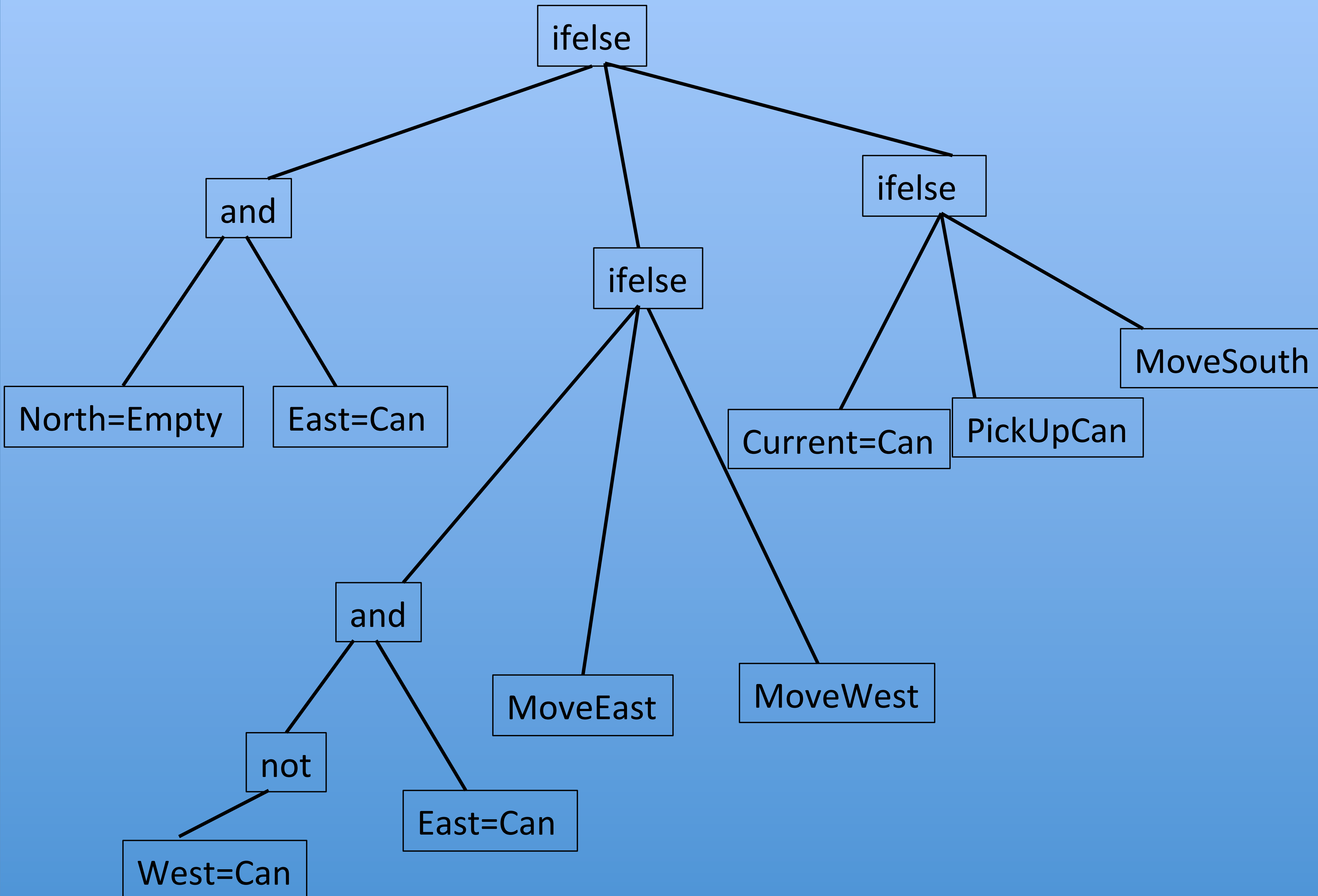
Genetic Programming
(John Koza, 1990s)

***Tree* representation of programs**



if (East=Can and North=Empty)
then MoveEast
else MoveSouth

A more complicated tree



Initial Population

Generate a population of *random trees*

Need to enforce some syntactic constraints, e.g., *ifelse* at root of tree, etc.

Fitness Calculation and Selection

Fitness:

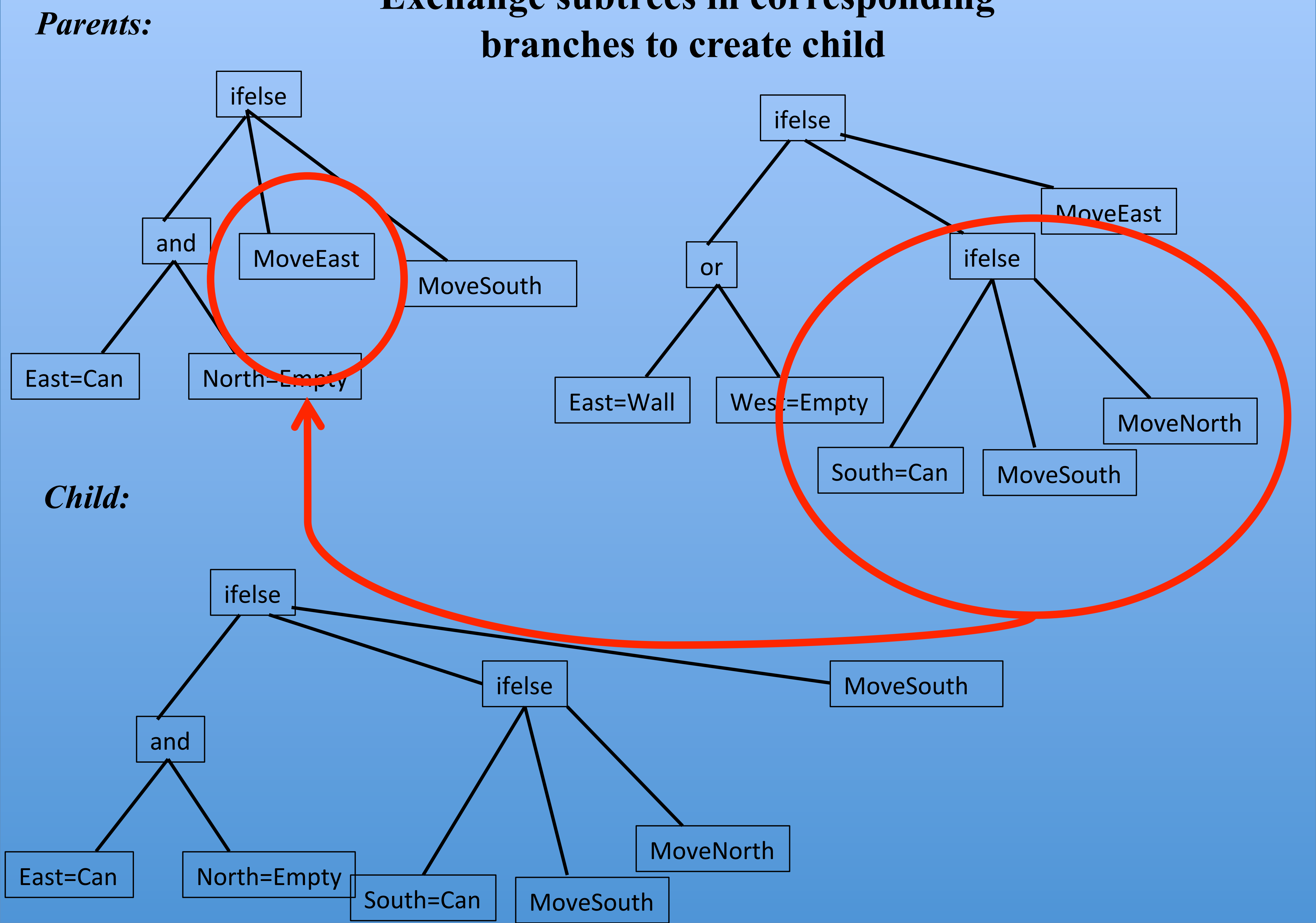
Have Robby try out each strategy in a variety of environments; compute each strategy's average score

Selection:

Fitter individuals create more offspring than less fit individuals

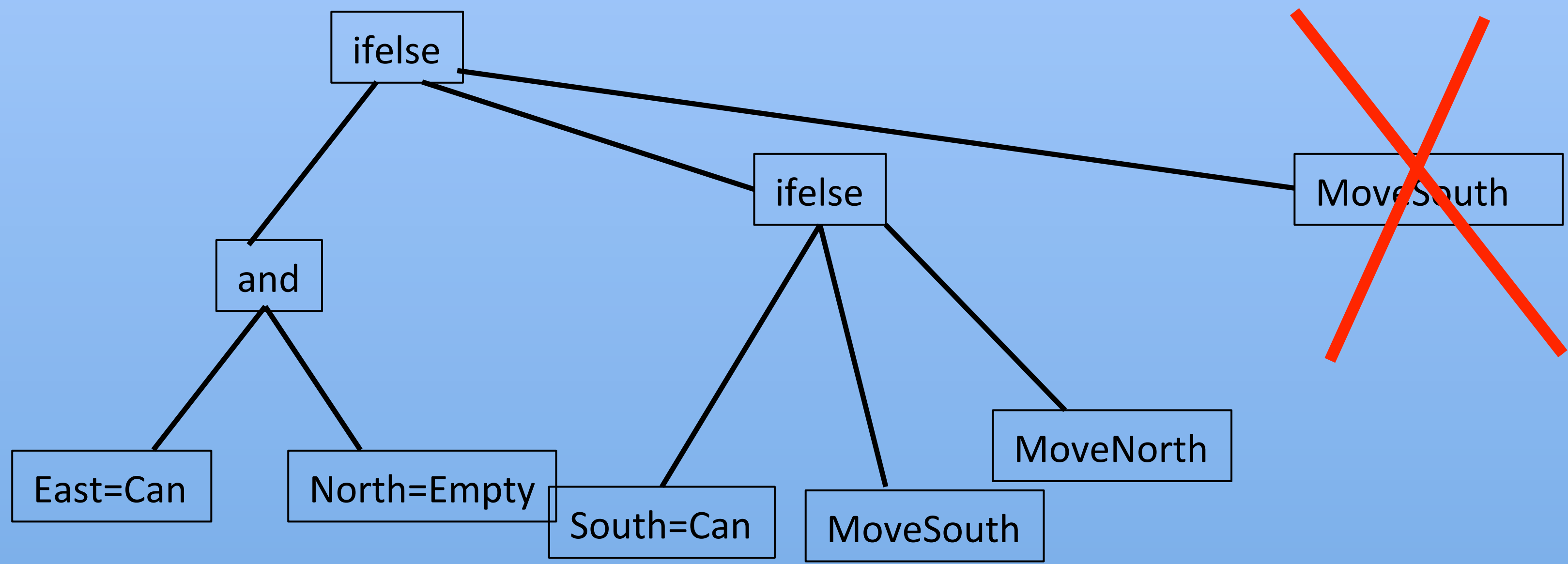
Crossover:

Exchange subtrees in corresponding branches to create child



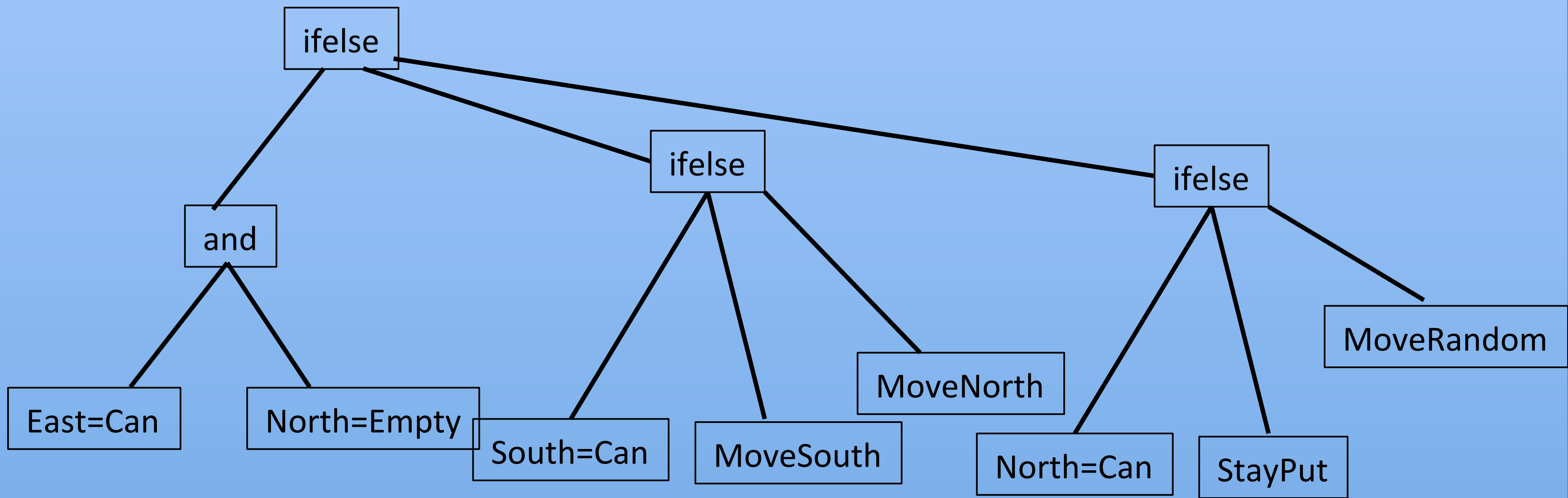
Mutation:

Replace a subtree by a randomly generated subtree



Mutation:

Replace a subtree by a randomly generated subtree



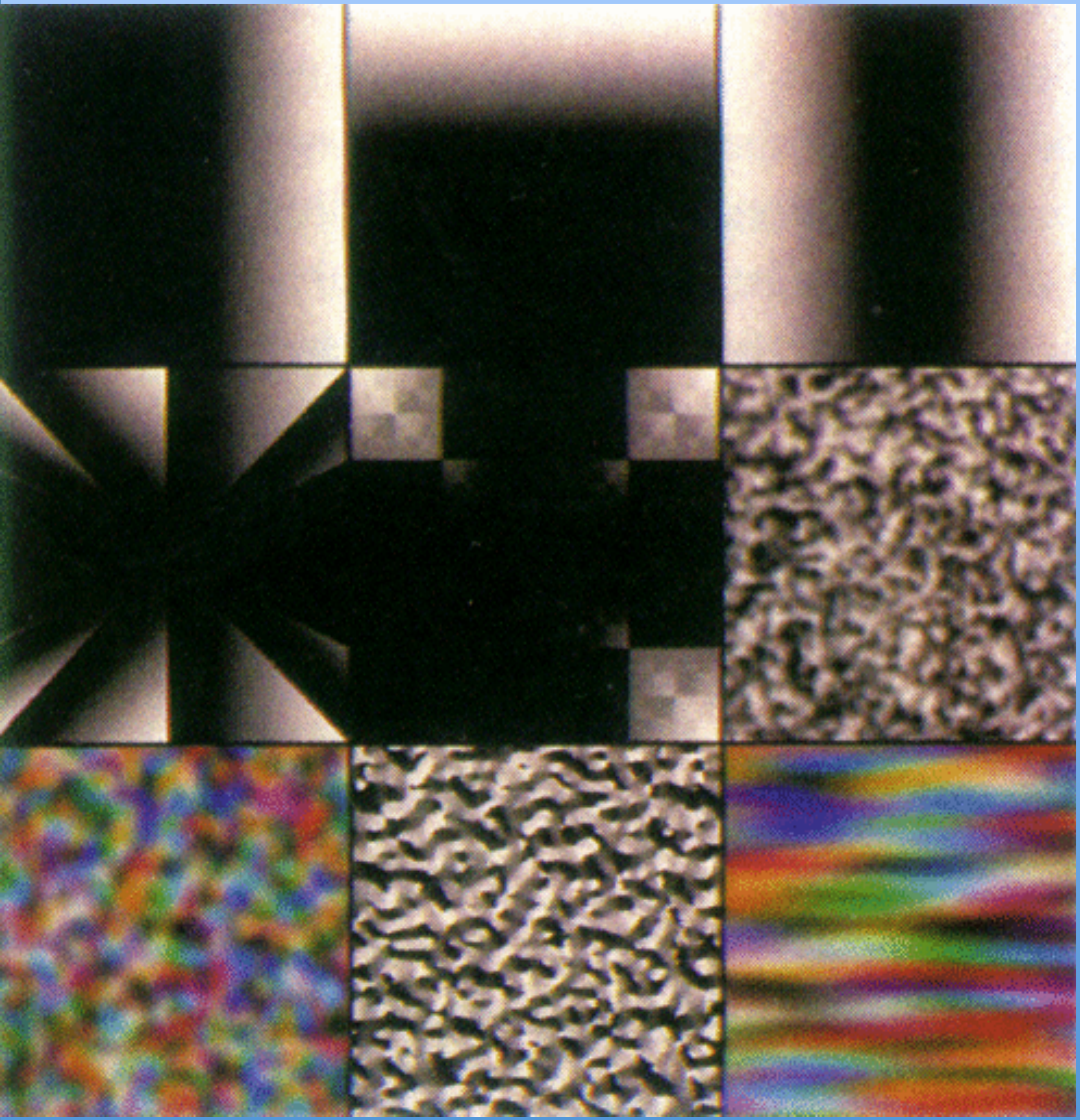
Genetic programming applied to Computer Graphics (Karl Sims, 1993)

Karl Sims

Genetic programming applied to Computer Graphics (Karl Sims, 1993)

- GA individuals: trees representing equations that generate a color for each pixel coordinate

Each function returns an image (an array of pixel colors)



Left to right, top to bottom:

a. X

b. Y

c. (abs X)

d. (mod X (abs Y))

e. (and X Y)

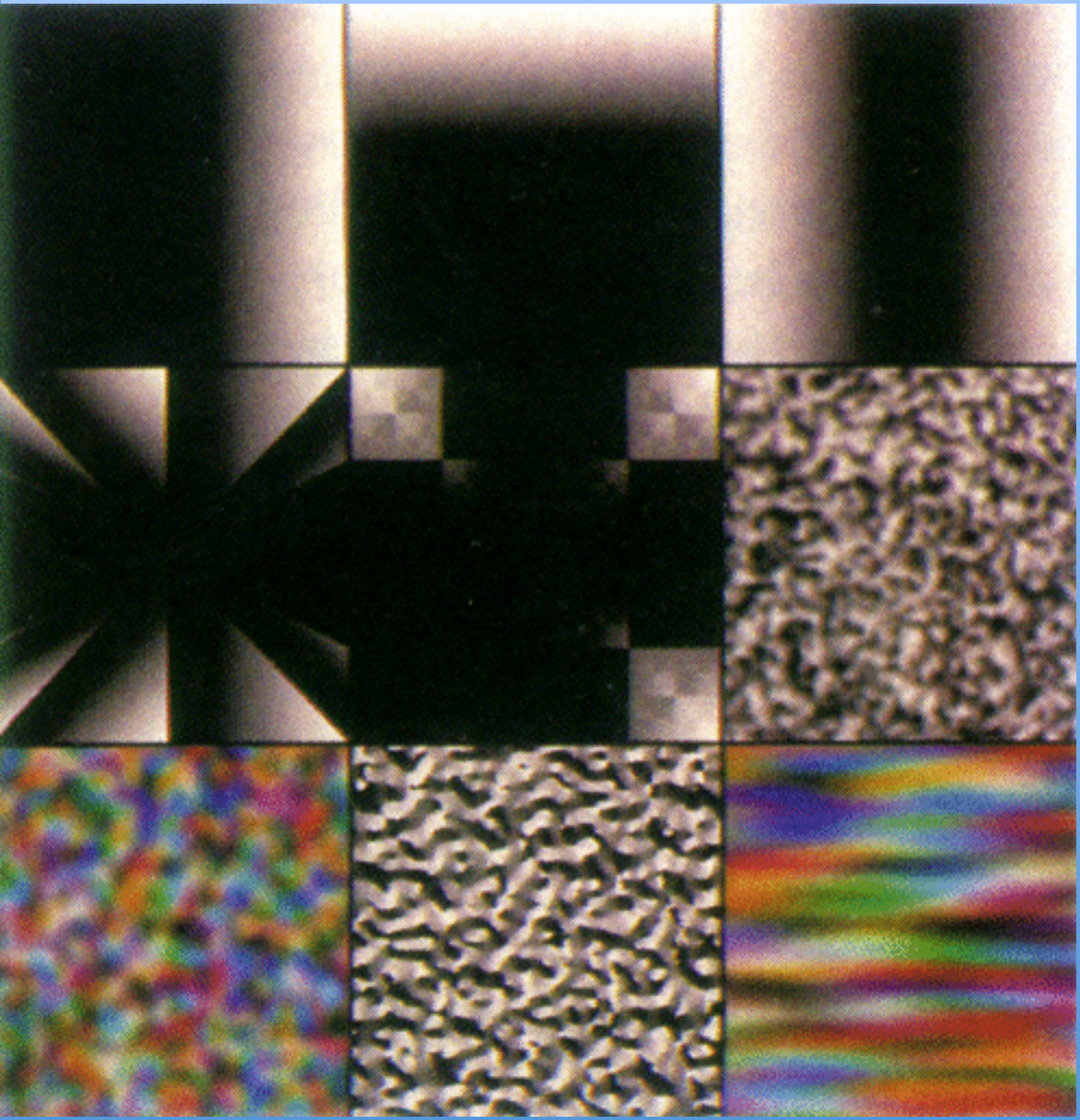
f. (bw-noise .2 2)

g. (color-noise .1 2)

h. (grad-direction
 (bw-noise .15 2)
 0 0)

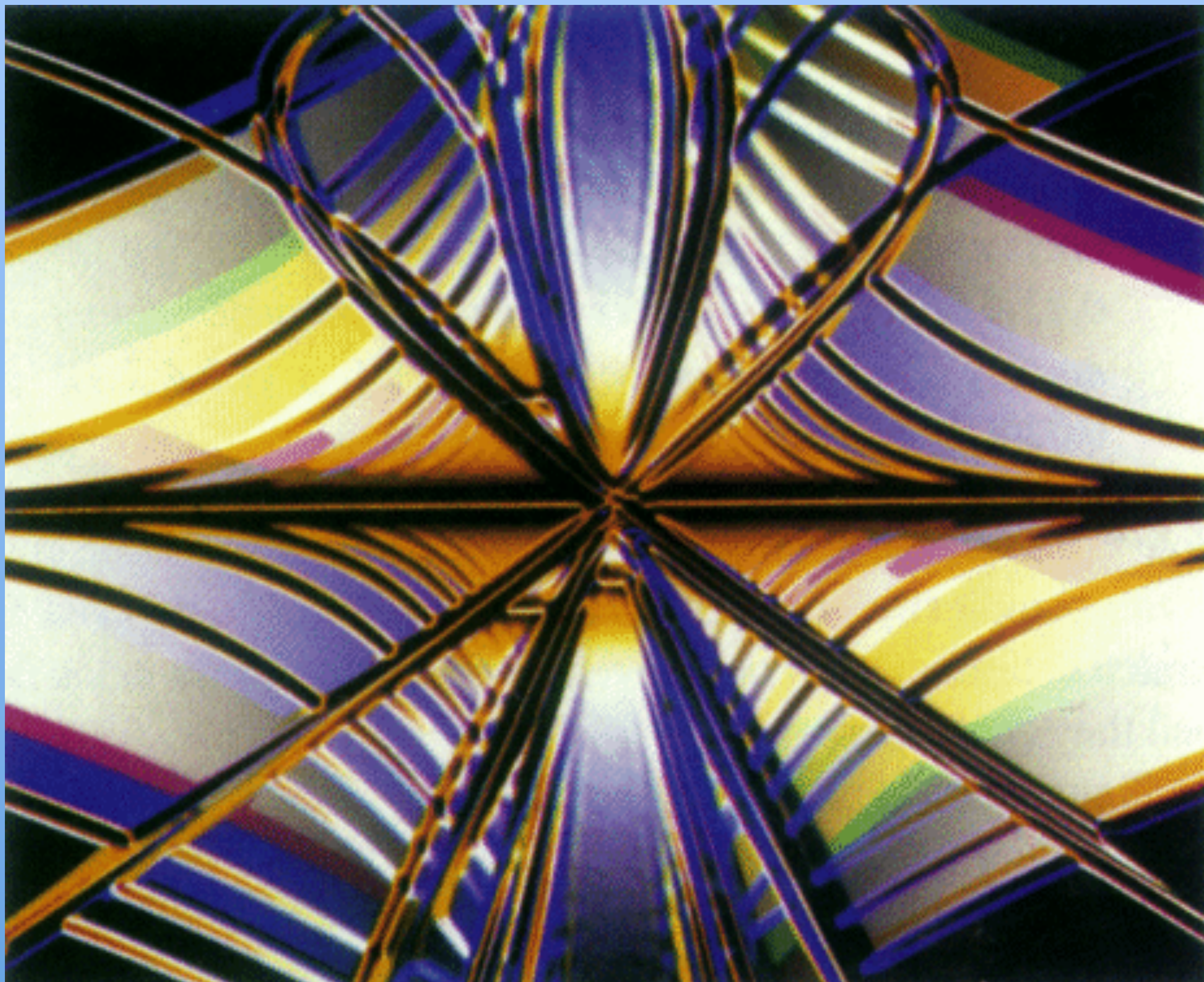
i. (warped-color-noise
 (* X .2) Y .1 2)

Each function returns an image (an array of pixel colors)

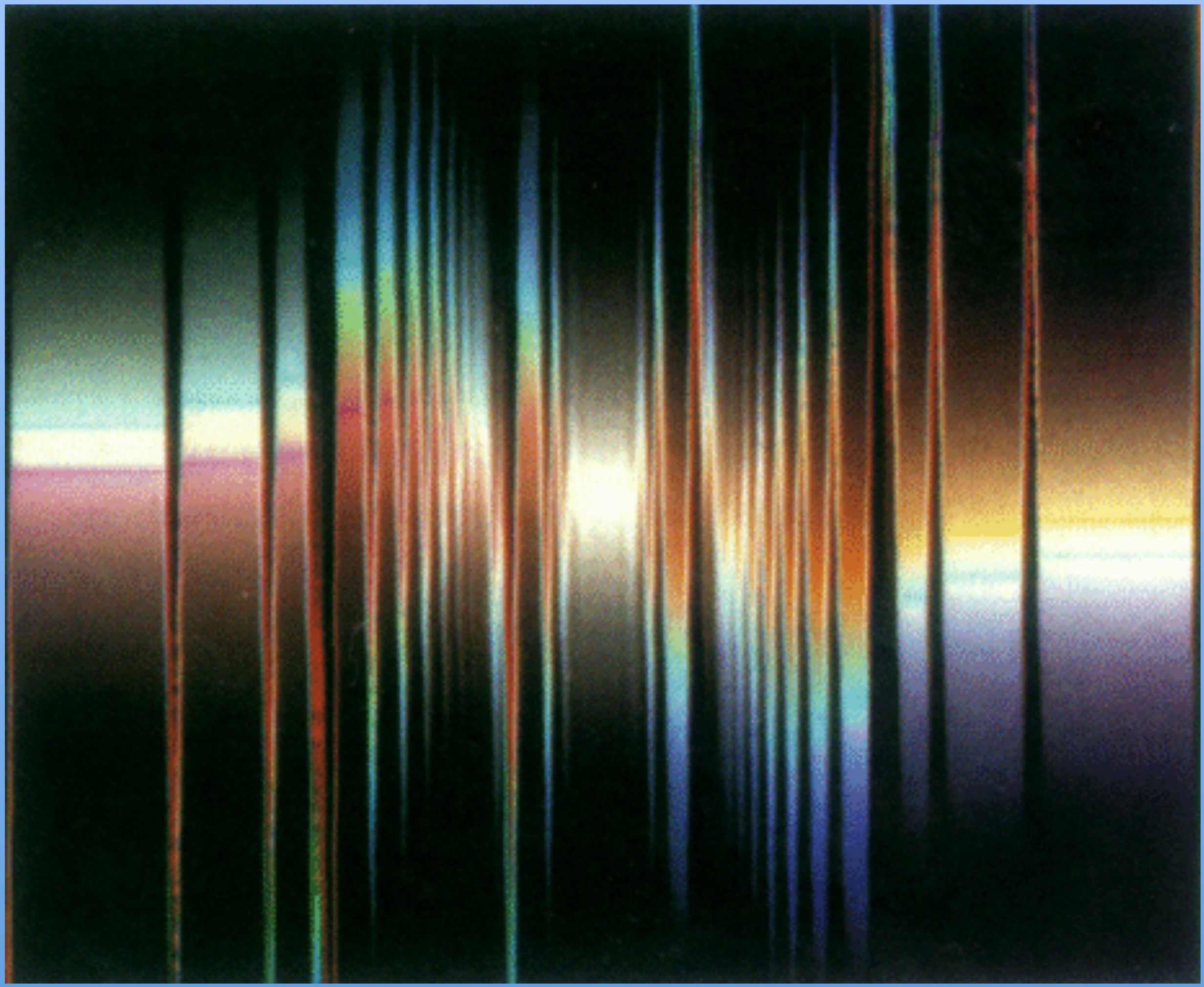


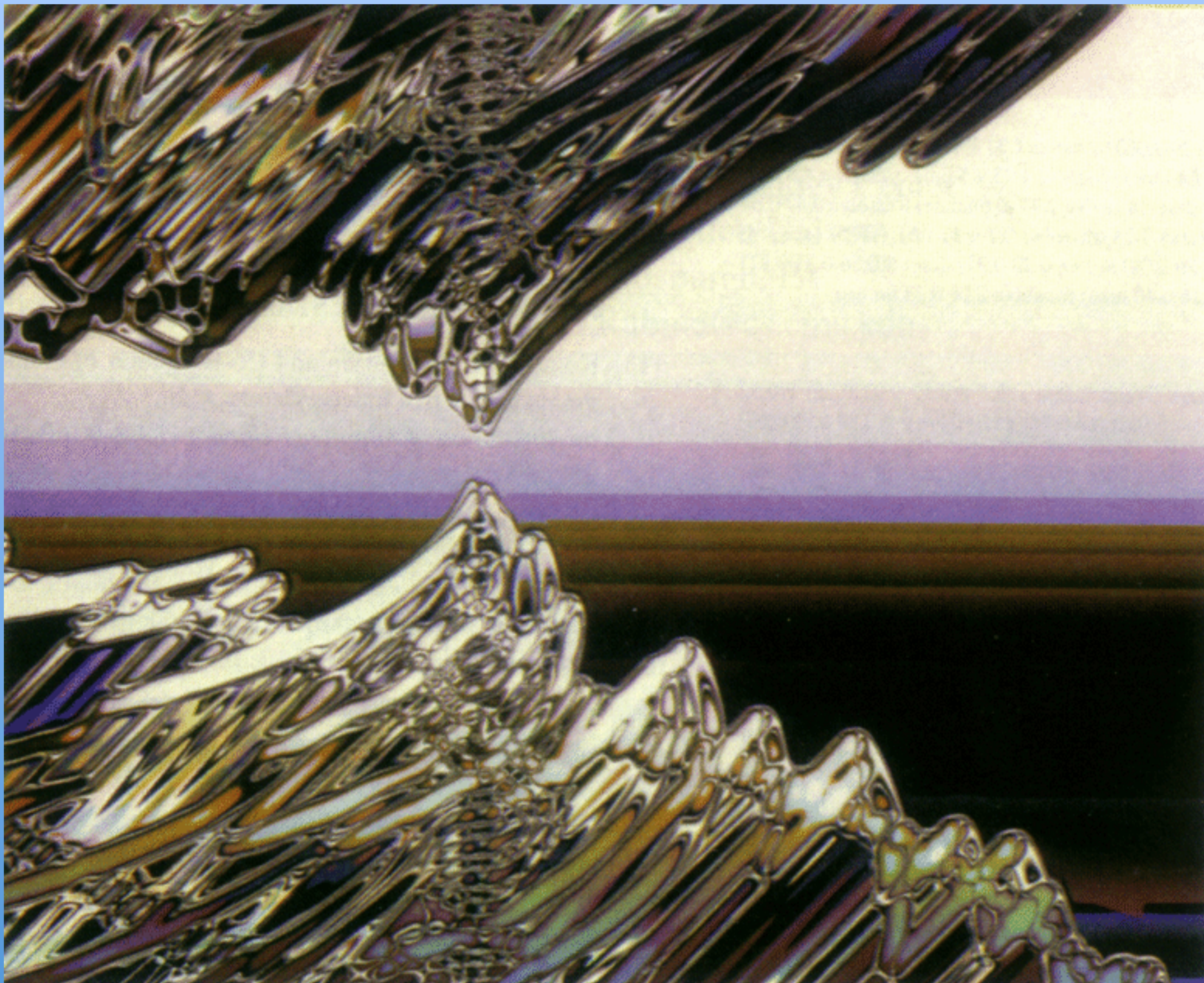
Fitness function???

Some Results



(round (log (+ y (color-grad (round (+ (abs (round (log (+ y (color-grad (round (+ y (log (invert y) 15.5)) x) 3.1 1.86 #(0.95 0.7 0.59) 1.35)) 0.19) x)) (log (invert y) 15.5)) x) 3.1 1.9 #(0.95 0.7 0.35) 1.35)) 0.19) x)





- **Website:** http://www.karlsims.com/

- **Applet:** http://www.jhlabs.com/java/art.html

http://www.virtualart.at/typo3temp/pics/ecc79d3cb6.jpg

From: http://www.karlsims.com/genetic-images.html

The viewers at this exhibit can observe a computer-simulated evolution in progress: an evolution of images. But in this evolution, the viewers are not just observers: they cause the evolution and direct its course.

From: http://www.karlsims.com/genetic-images.html

A population of images is displayed by the computer on an arc of 16 video screens. The viewers determine which images will survive by standing on sensors in front of those they think are the most aesthetically interesting. The pictures that are not selected are removed and replaced by offspring from the surviving images. The new images are copies and combinations of their parents, but with various alterations. This is an artificial evolution in which the viewers themselves interactively determine the "fitness" of the pictures by choosing where they stand. As this cycle continues, the population of images can progress towards more and more interesting visual effects.

From: http://www.karlsims.com/genetic-images.html

This interactive installation is an unusual collaboration between humans and machine: the humans supply decisions of visual aesthetics, and the computer supplies the mathematical ability for generating, mating, and mutating complex textures and patterns. The viewers are not required to understand the technical equations involved. The computer can only experiment at random with no sense of aesthetics — but the combination of human and machine abilities permits the creation of results that neither of the two could produce alone.

From: http://www.karlsims.com/genetic-images.html

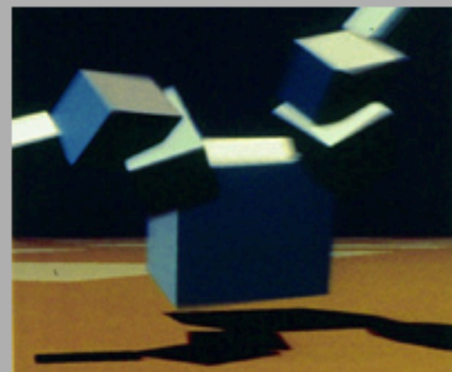
A population of images is displayed by the computer on an arc of 16 video screens. The viewers determine which images will survive by standing on sensors in front of those they think are the most aesthetically interesting. The pictures that are not selected are removed and replaced by offspring from the surviving images. The new images are copies and combinations of their parents, but with various alterations. This is an artificial evolution in which the viewers themselves interactively determine the "fitness" of the pictures by choosing where they stand. As this cycle continues, the population of images can progress towards more and more interesting visual effects.

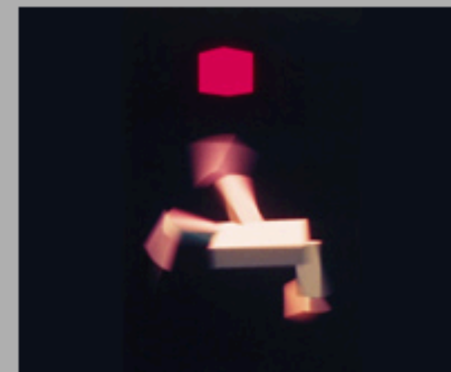
The viewers of this simulated evolution collectively determine its pathway to previously unseen populations of pictures.

This interactive installation is an unusual collaboration between humans and machine: the humans supply decisions of visual aesthetics, and the computer supplies the mathematical ability for generating, mating, and mutating complex textures and patterns. The viewers are not required to understand the technical equations involved. The computer can only experiment at random with no sense of aesthetics — but the combination of human and machine abilities permits the creation of results that neither of the two could produce alone.

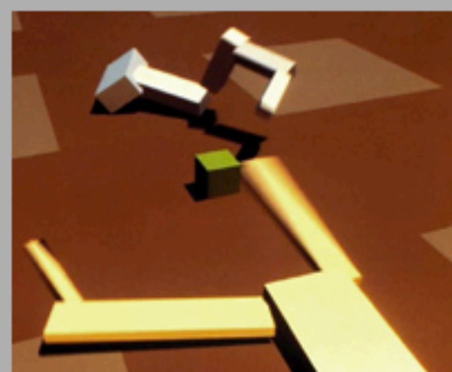
Evolving Virtual Creatures
Sims, 1994


Swimming


Hopping


Following


Competing



http://www.karlsims.com/evolved-virtual-creatures.html

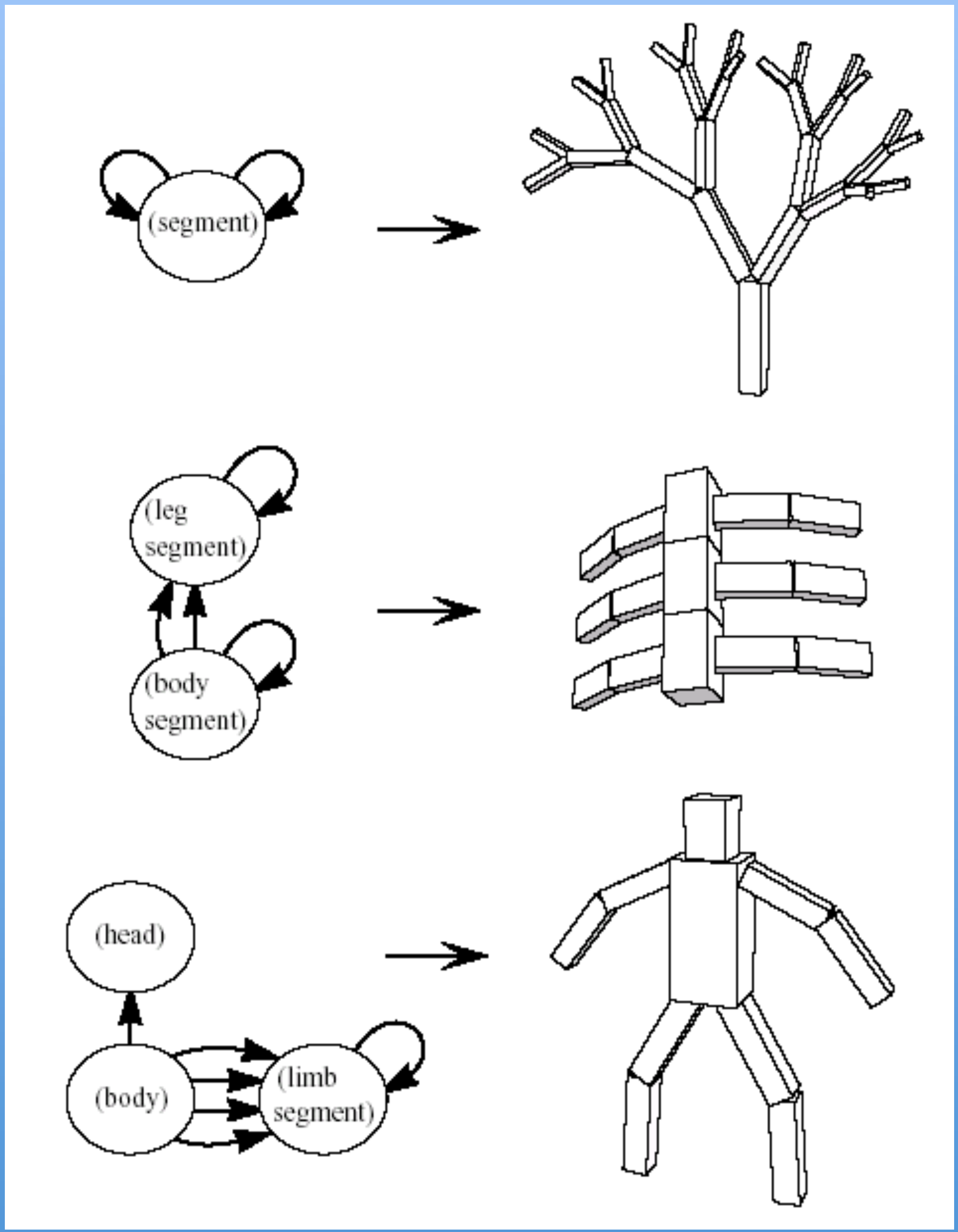
Evolving Virtual Creatures
Sims, 1994

- Goal is to evolve life-like simulated "creatures".
- Simultaneously evolve morphology and control systems (neural networks) of simulated creatures that can "walk", swim, jump, follow light.
- Environment is detailed simulation of physics.

Evolving virtual creatures: Some details

(optional)

The creatures' body morphologies were represented as "grammars" and evolved.



Genotype graphs and corresponding morphologies

Creatures' "brains"

Neural network controlling each body part:

- Function that inputs sensor values and outputs effector values (forces or torques at joints)

Sensors:

- **Joint-angle sensors**: returns value of angle
- **Contact sensors:** 1.0 if contact is made (with anything), -1.0 if no contact
- **Photosensors:** return coordinates of light source direction relative to orientation of part

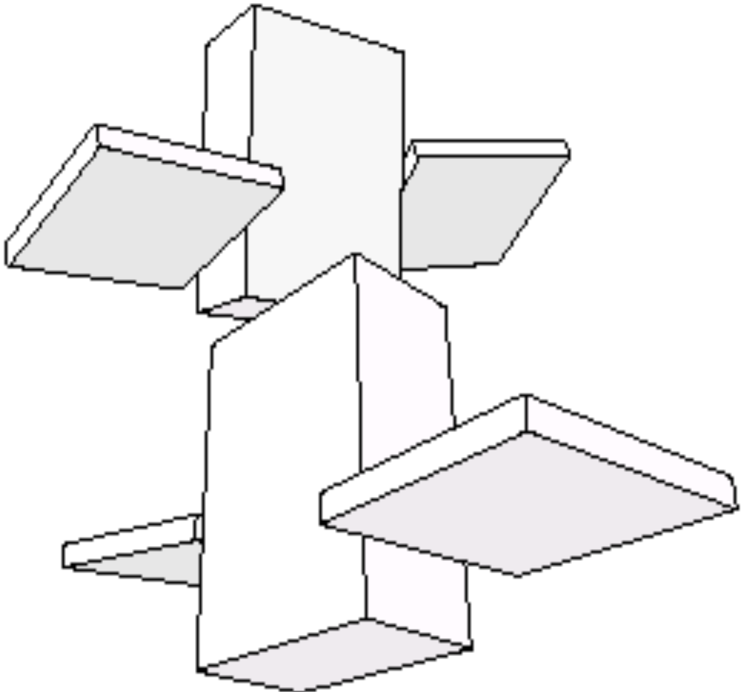
Creatures' "brains"

- **Neurons:** At each simulated time step, every neuron computes its output value from its inputs.

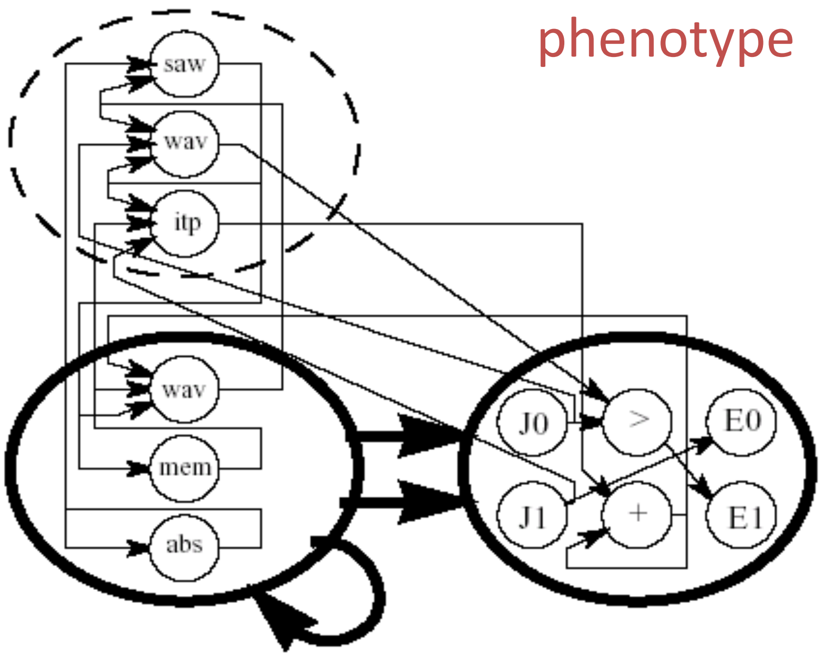
- **Effectors ("muscles"):**

  - Each effector controls a degree of freedom of a joint.

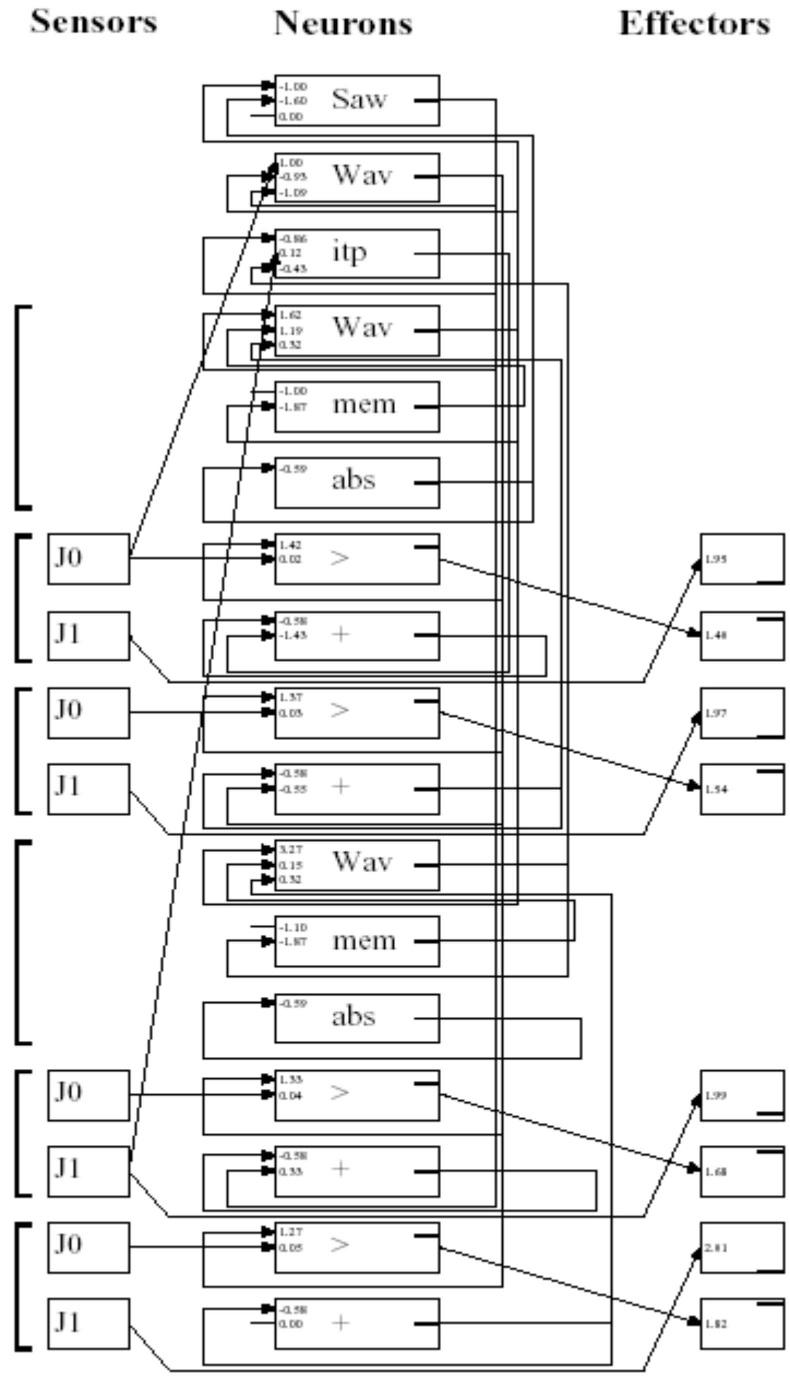
  - Each effector gets input from a single neuron or sensor, and outputs joint force



Body phenotype



Genotype



Neural network phenotype

Physical Simulation

- **Complicated physical simulation is used:**
  - articulated body dynamics, numerical integration, collision detection, collision response, friction, viscous fluid effect.
- **Parallel implementation on a Connection Machine**

Evolution of Creatures

- **Creature grown from its genetic description.**
- **Run in simulation for some period of time**
  - Sensors provide data about world and creature's body to brain
  - Brain produces effector forces which move parts of creature
  - Fitness: how successful was desired behavior by end of allotted time