Biologically Inspired AI

Some Examples of Biologically Inspired AI

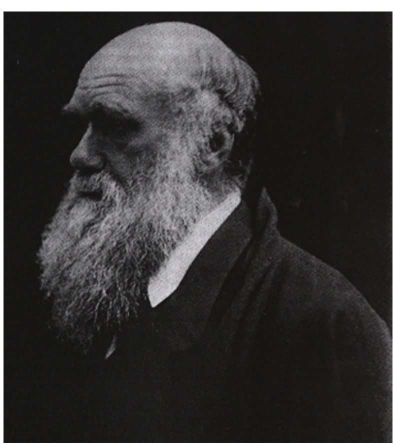
- Neural networks
- Evolutionary computation (e.g., genetic algorithms)
- Immune-system-inspired computer/network security
- Ant-colony optimization
- Swarm intelligence (e.g., decentralized robots)
- Molecular (DNA) computation

Evolutionary Computation

A collection of computational methods inspired by biological evolution:

- A population of candidate solutions evolves over time, with the fittest at each generation contributing the most offspring to the next generation
- Offspring are produced via crossover between parents, along with random mutations and other "genetic" operations.

Evolution made simple



Charles Darwin
1809–1882

Essentials of Darwinian evolution:

- Organisms reproduce in proportion to their *fitness* in the environment
- Offspring inherit traits from parents
- Traits are inherited with some variation, via mutation and sexual recombination

Evolution made simple

Essentials of evolutionary algorithms:

- Computer "organisms" (e.g., programs) reproduce in proportion to their *fitness* in the environment (e.g., how well they perform a desired task)
- Offspring inherit traits from their parents
- Traits are inherited, with some variation, via mutation and "sexual recombination"

Essentials of Darwinian evolution:

- Organisms reproduce in proportion to their *fitness* in the environment
- Offspring inherit traits from parents
- Traits are inherited with some variation, via mutation and sexual recombination

Appeal of ideas from evolution:

- Successful method of searching large spaces for good solutions (chromosomes / organisms)
- Massive parallelism
- Adaptation to environments, change
- Emergent complexity from simple rules

Genetic Algorithms

Components of a GA:

- *Population* of candidate solutions to a given problem ("chromosomes")
- *Fitness function* that assigns fitness to each chromosome in the population
- *Selection procedure* that selects individuals to reproduce
- *Genetic operators* that take existing chromosomes and produce offspring with variation (e.g., mutation, crossover)

A Simple Genetic Algorithm

1. Start out with a randomly generated population of chromosomes (candidate solutions).
2. Calculate the fitness of each chromosome in the population.
3. Select pairs of parents with probability a function of fitness rank in the population.
4. Create new population: Cross over parents, mutate offspring, place in new population.
5. Go to step 2.

Genetic operators

- *Crossover:* exchange subparts of two chromosomes:

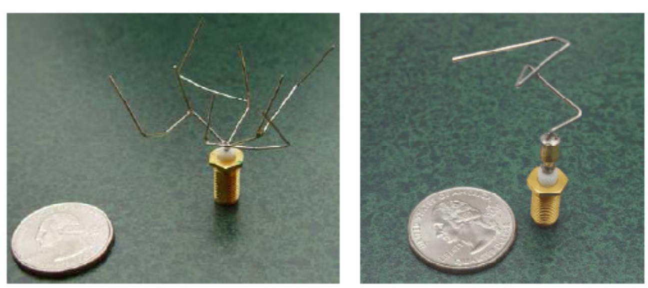
0 0 0 | 0 0 1 0 0 0 → 0 0 0 1 1 1 1 1 1

0 0 1 | 1 1 1 1 1 1 → 0 0 1 0 0 1 0 0 0

- *Mutation:* randomly change some loci:

0 0 0 0 0 1 0 0 0 → 0 0 0 0 0 0 0 0 0

From Hornby et al., 2006

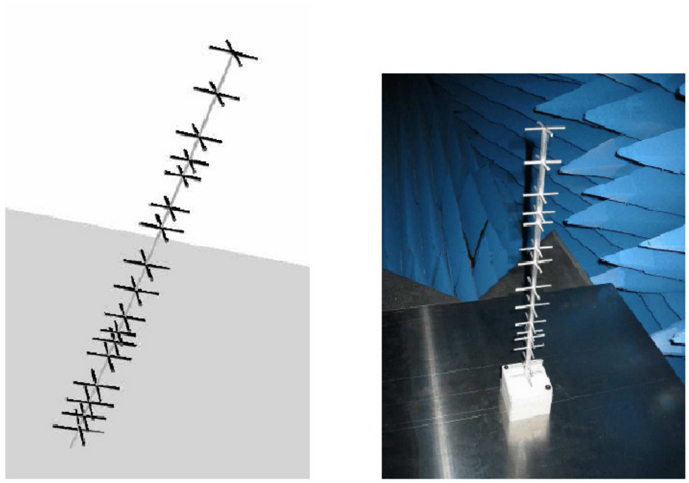


(a) (b)

Figure 2. Photographs of prototype evolved antennas: (a) the best evolved antenna for the initial gain pattern requirement, ST5-3-10; (b) the best evolved antenna for the revised specifications, ST5-33-142-7.

Evolvable hardware work at NASA Ames
(Hornby, Lohn, et al.)

From Hornby et al., 2006



(a) (b)

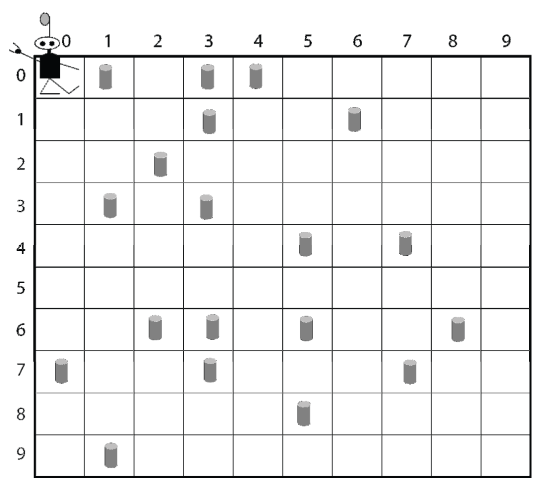
Figure 4. Best evolved TDRS-C antenna: (a) simulation and (b) fabricated.

Example: Evolving Strategies for Robby the Robot

Sensors:
N,S,E,W,C(urrent)

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 fitness function

```
Total_Reward = 0;
Average_Reward = 0
For i = 1 to NUM_ENVIRONMENTS {
  generate_random_environment( );
  For j = 1 to NUM_MOVES_PER_ENVIRONMENT {
      Total_Reward = Total_Reward +
                     perform_action( );
  }
}

Fitness = Total_Reward / NUM_ENVIRONMENTS;
```

Robby's Representation

Look-up table: state → action

States:

Contents of [N, S, E, W, C]

Contents of each direction (and C) is represented by two bits.

Empty = 00
Can = 01
Wall = 10

Actions: MoveN = 0; MoveS = 1; MoveE = 2; MoveW = 3;
StayPut = 4; MoveRandom = 5; TryToPickUpCan = 6

Example of look-up table:

N	S	E	W	C	Action
00	00	00	00	00	1
00	00	00	00	01	3
00	00	00	00	10	5
00	00	00	00	11	2
00	00	00	01	00	6
		• • •			
11	11	11	11	11	4

How many lines in the table? How many possible "strategies"? What about the non-coding genes?

Example of look-up table:

N	S	E	W	C	Action
00	00	00	00	00	1
00	00	00	00	01	3
00	00	00	00	10	5
00	00	00	00	11	2
00	00	00	01	00	6
		• • •			
11	11	11	11	11	4

Robby's strategy chromosome: 1 3 5 2 6 ... 4

GA for evolving strategies for Robbie

For 1000 generations:

1. Generate 200 random strategies

2. For each strategy, calculate fitness (average points earned on 100 random environments)

3. Rank the population in order of decreasing fitness

4. Copy the 30 highest-fitness individuals without change to the next generation.

5. From these 30 highest individuals, choose pairs at random to be parents.

6. Perform crossover and mutation to form two offspring from each pair and add offspring to next generation. Continue until the next generation has 200 individuals.

7. Go to step 2.

Results from one run