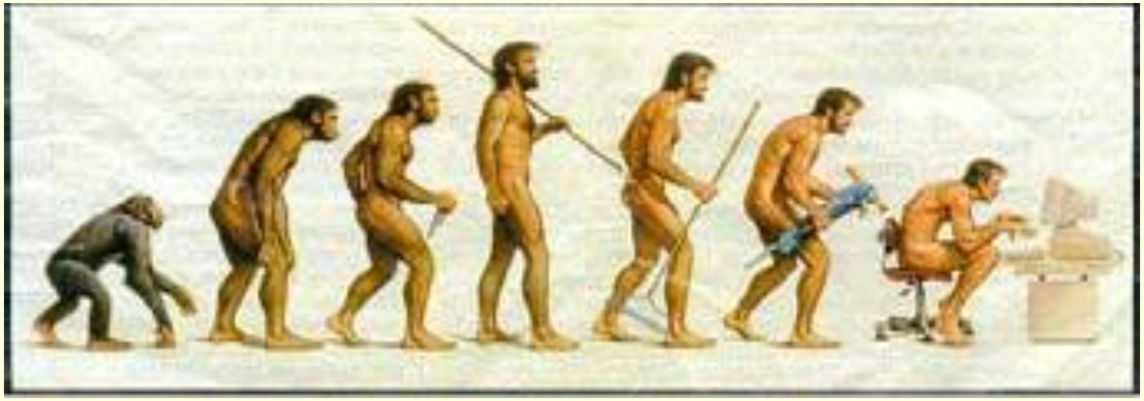
Genetic algorithms

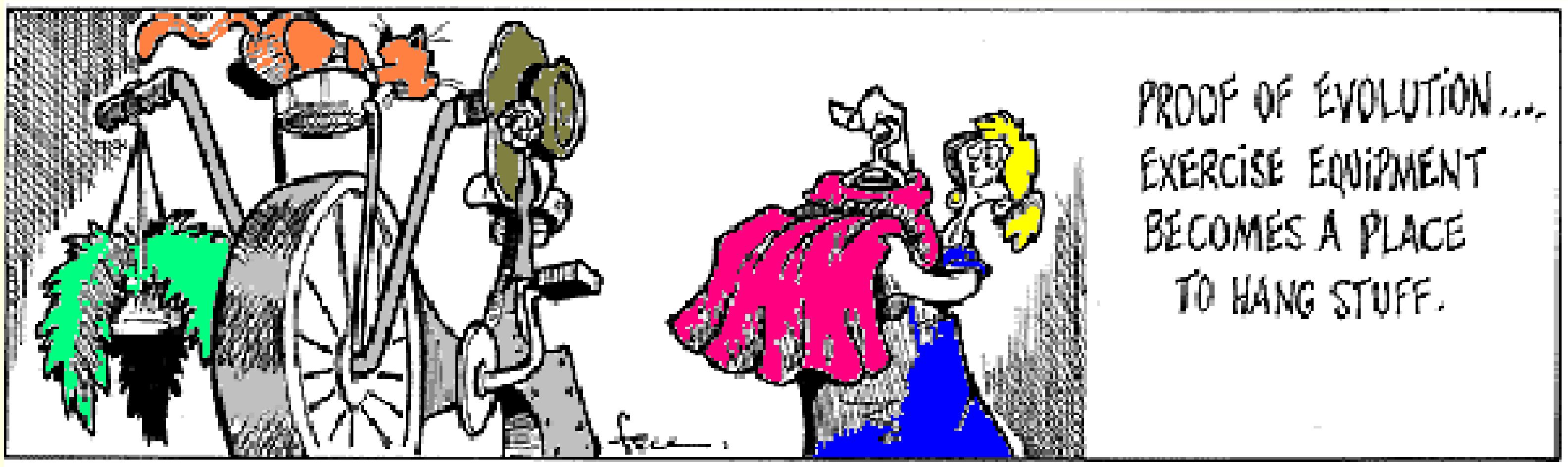
by Richard Simpson



TEXT: An Introduction to Genetic Algorithms by Melanie Mitchell

Evolution?

- What is evolutionary computation?
- Why is this an interesting approach?



Genetic Algorithms

- Premise
  - Evolution works biologically so maybe it will work with simulated environments.
  - Here each possible solution (good or bad) is represented by a chromosome (ie a pattern of bits which is sort of synonymous to DNA)
  - Determine the better solutions
  - Mate these solutions to produce new solutions which are (hopefully!) occasionally better than the parents.
  - Do this for many generations.

Originator

- John Holland and colleagues at the University of Michigan
  - Developed GA’s during the 70’s
  - Theoretical until the mid-1980s
  - Originally was used to study the behavior of evolutionary systems in nature.
- Seminal work
  - Adaptation in Natural and Artificial Systems introduced main GA concepts, 1975

Introduction

- Computing pioneers looked to natural systems as guiding metaphors
- Evolutionary computation
  - Any biologically-motivated computing activity simulating natural evolution
- Genetic Algorithms are one form of this activity
- Genetic Programming is another

Main idea

- Take a population of candidate solutions to a given problem
- Use operators inspired by the mechanisms of natural genetic variation
- Apply selective pressure toward certain properties
- Evolve a more fit solution

Why evolution as a metaphor

- Ability to efficiently guide a search through a large solution space
- Ability to adapt solutions to changing environments
- “Emergent” behavior is the goal
  - “The hoped-for emergent behavior is the design of high-quality solutions to difficult problems and the ability to adapt these solutions in the face of a changing environment”
    - Melanie Mitchell, An Introduction to Genetic Algorithms

Evolutionary terminology

- Abstractions imported from biology
  - Chromosomes, Genes, Alleles
  - Fitness, Selection
  - Crossover, Mutation

GA terminology

- In the spirit – but not the letter – of biology
  - GA chromosomes are strings of genes
    - Each gene has a number of alleles; i.e., settings
  - Each chromosome is an encoding of a solution to a problem
  - A population of such chromosomes is operated on by a GA

Encoding

- A data structure for representing candidate solutions
  - Often takes the form of a bit string, array of integers or some other data type. May even be a tree.
  - Usually has internal structure; i.e., different parts of the string represent different aspects of the solution)

Crossover

- Mimics biological recombination
  - Some portion of genetic material is swapped between chromosomes
  - Typically the swapping produces an offspring
- Mechanism for the dissemination of “building blocks” (schemas)

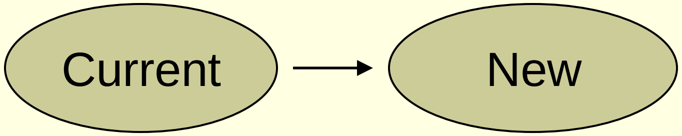
Mutation

- Selects a random locus – gene location – with some probability and alters the allele at that locus
- The intuitive mechanism for the preservation of variety in the population

Fitness

- A measure of the goodness of the organism
- Expressed as the probability that the organism will live another cycle (generation)
- Basis for the natural selection simulation
  - Organisms are selected to mate with probabilities proportional to their fitness
- Probabilistically better solutions have a better chance of conferring their building blocks to the next generation (cycle)

A Simple GA




```
  Generate initial population (current)
  do
      Calculate the fitness of each member
      do
          Select parents from current population
          where the probability of selection is an
          increasing function of fitness.
          Perform crossover and add offspring to
          the new population
          Mutate the offspring
      while new population is not full
      Replace current population with the new
while not converged
```

How do GAs work

- The structure of a GA is relatively simple to comprehend, but the dynamic behavior is complex
- Holland has done significant work on the theoretical foundations of GAs
- “GAs work by discovering, emphasizing, and recombining good ‘building blocks’ of solutions in a highly parallel fashion.”
  - Melanie Mitchell, paraphrasing John Holland

Lets Look at a simple example

- Suppose that we have a string of bits say 16 bits long.
- We would like to create a string of bits that have only 1 bits using evolutionary methods
- First we create a random population of 16 bit strings. Let popsize=100
- We then define a fitness function f(s) that counts the number 1 bits in the string and returns that number.

Fitness Proportionate selection

- In this case the number of times an individual is expected to reproduce is equal to its fitness divided by the average of fitnesses in the populations.
- Roulette-wheel sampling is how we implement the above. This is conceptually equivalent to giving each individual a slice of a circular roulette wheel equal in area to the individual's fitness.

Roulette Wheel sampling.

Suppose we are dealing with 8 bit strings and we have the following population and associated fitnesses.

10110101	5	17
00001111	4	12
10101010	4	8
10001001	3	4
00000000	0	0
01000000	1	1

Cumulative distribution

Let S= Sum of fitnesses

Select random real number between 0 and S.

Compare value to the sequence of partial sums to select individual.

Higher fitness individuals have higher probability of selection.

Crossover Operator

- There are many ways to do crossovers. One of the simplest is call the single point crossover. Select pos. and swap.

Example Fit

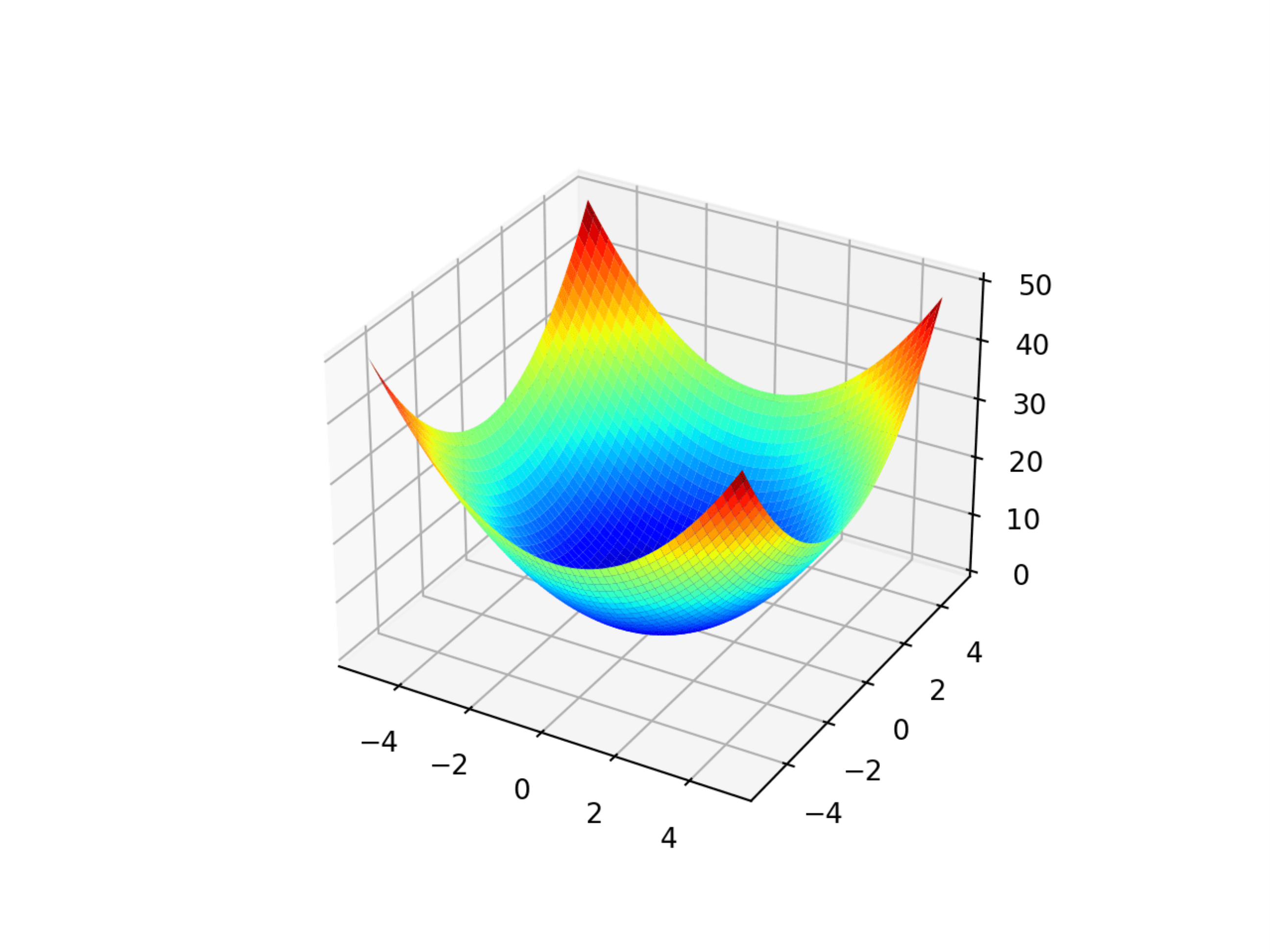
P1 = 1 0 1 1 0 0 1 0 0 0	0 0 0 0 1 1	6
P2 = 1 0 1 1 1 1 0 1 0 0	1 1 1 0 0 0	9
O1 = 1 0 1 1 0 0 1 0 0 0	1 1 1 0 0 0	7
O2 = 1 0 1 1 1 1 0 1 0 0	0 0 0 0 1 1	8

SO!

```
Generate 100 random strings 16 bits long.
do
      Fitness = sum of bits in each string
      do
         Select parents from current population
            using roulette wheel selection
         Perform the discussed crossover
         Mutate the offspring with low probability
         Add children to new population
      while new population is not full
      Replace current population with the new
while not converged, whatever this means.
```

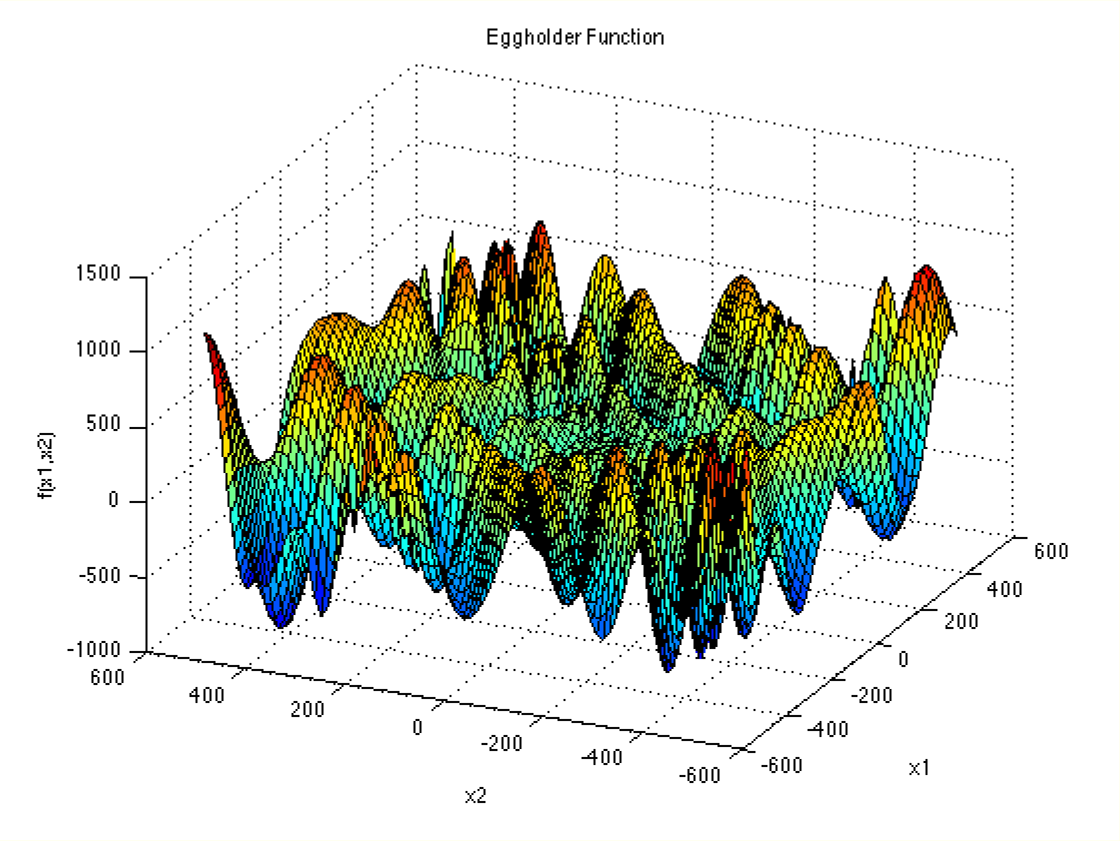
This will converge to a string of one's quite rapidly.

Simple landscape problem
Easily solved by GA



Example multimodal function

Needs species and crowding methods



Multimodal functions

https://en.wikipedia.org/wiki/Test_functions_for_optimization

https://www.sfu.ca/~ssurjano/optimization.html