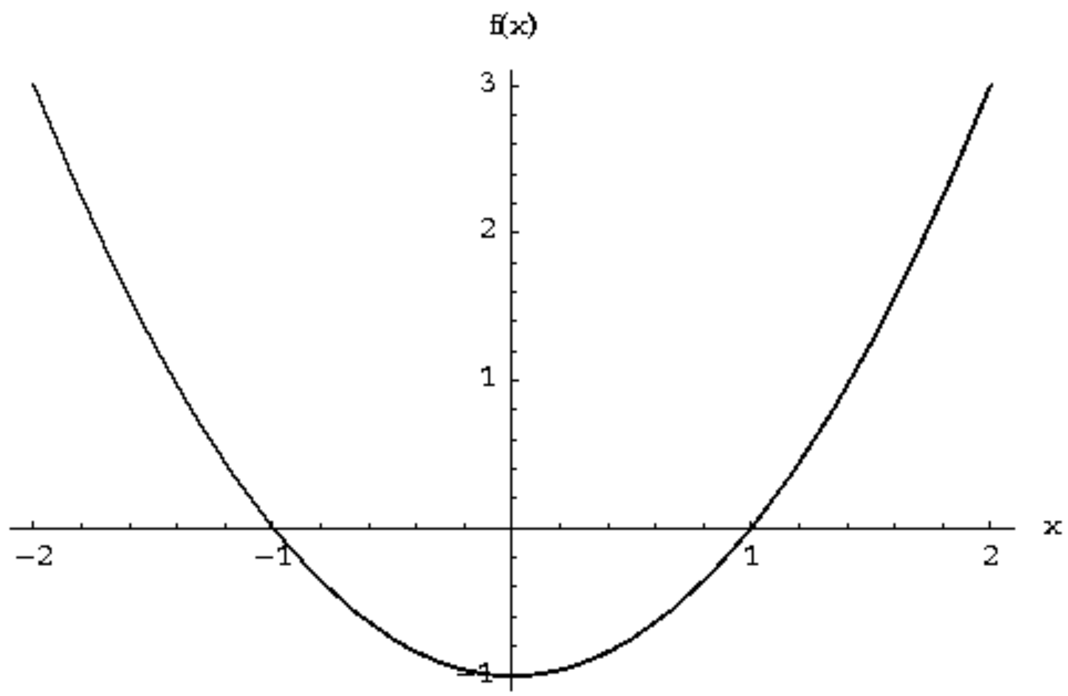
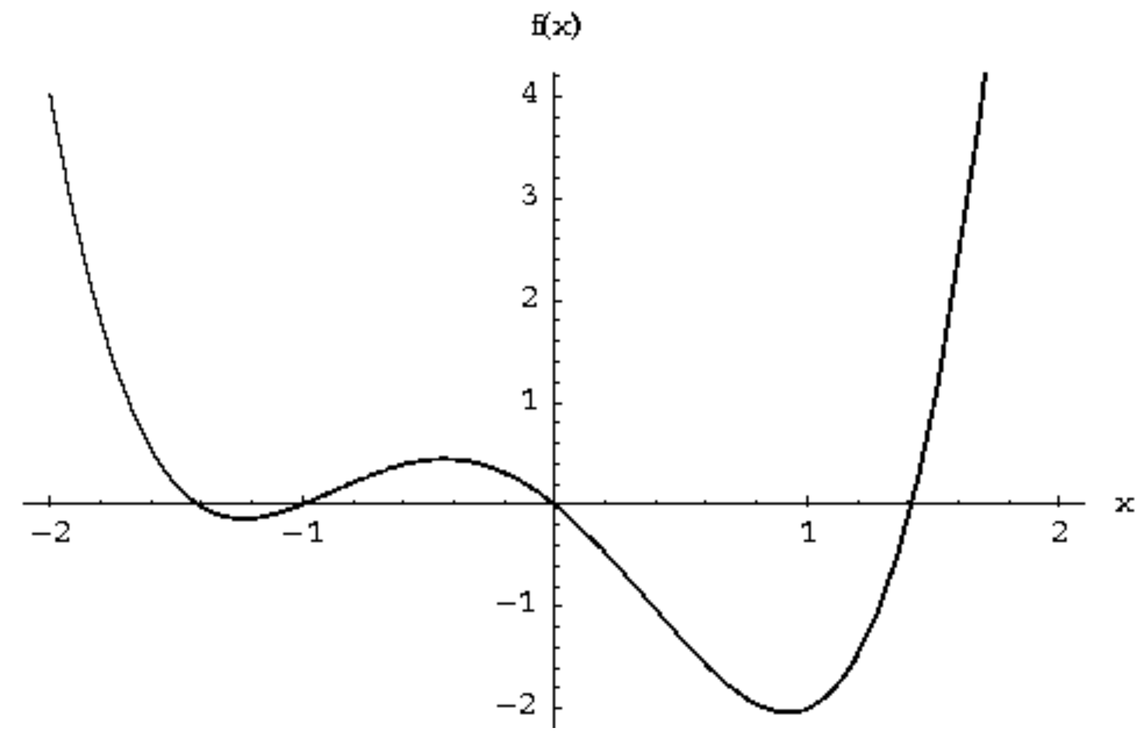
CARMINE-EMANUELE CELLA

INTRODUCTION TO GENETIC ALGORITHMS

MUSIC 159

Convex vs non-convex





A biology lesson

Every organism has a set of rules describing how that organism is built up from the tiny building blocks of life.

These rules are encoded in the genes of an organism, which in turn are connected together into long strings called **chromosomes**.

Genetic Algorithms are a way of solving problems by mimicking the same processes mother nature uses.

They use a combination of **selection**, **recombination** and **mutation** to evolve a solution to a problem.

Algorithm

At the beginning of a run of a genetic algorithm a large population of *random* chromosomes is created. Each one, will represent a different solution to the problem.

Then, the following steps are repeated until a solution is found:

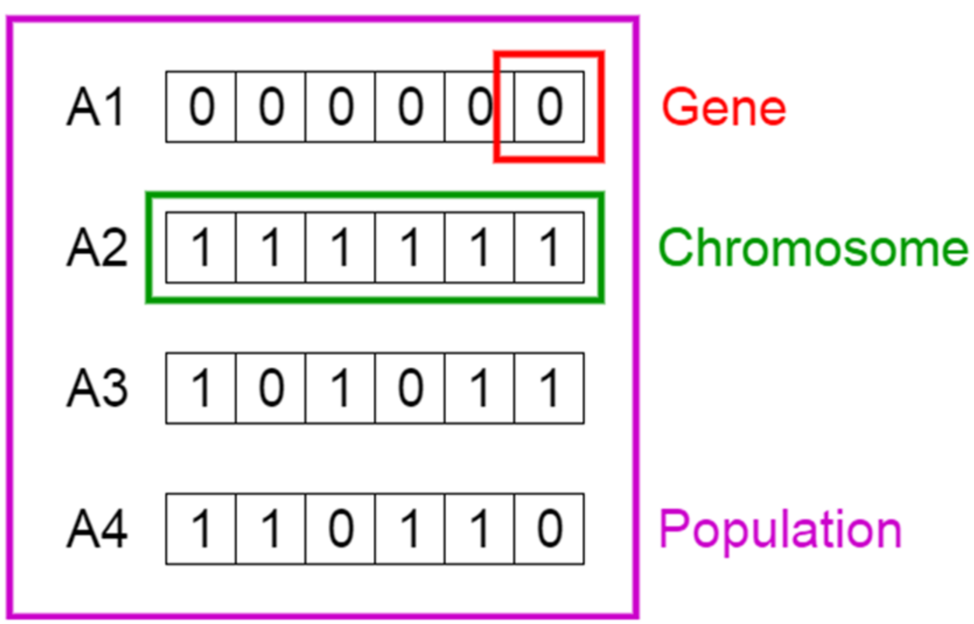
1. Test each chromosome to see how good it is at solving the problem and assign a **fitness** score accordingly.

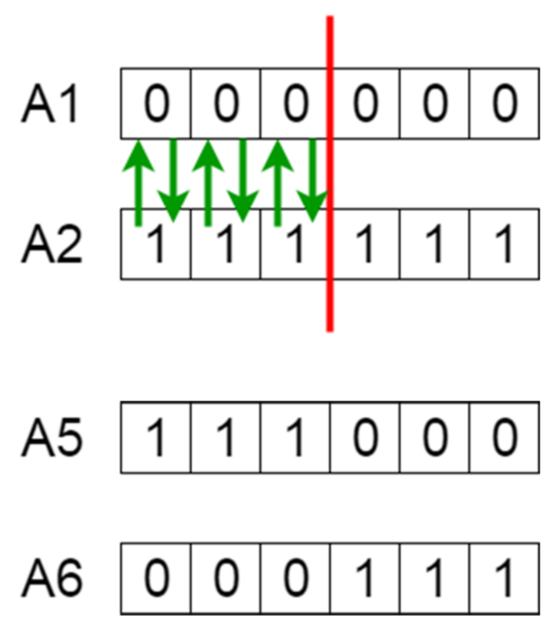
2. Select **two** members from the current population. The chance of being selected is proportional to the chromosomes fitness. *Roulette wheel* selection is a commonly used method.
3. Dependent on the **crossover** rate crossover the bits from each chosen chromosome at a randomly chosen point.
4. Step through the chosen chromosomes bits and flip dependent on the **mutation** rate.
5. Repeat step 2, 3, 4 until a new population of N members has been created.

Overview

Psuedocode

```
START
Generate the initial population
Compute fitness
REPEAT
    Selection
    Crossover
    Mutation
    Compute fitness
UNTIL population has converged
STOP
```





Crossover rate

This is simply the chance that two chromosomes will swap their bits. A good value for this is around 0.7. Crossover is performed by selecting a random gene along the length of the chromosomes and swapping all the genes after that point.

e.g. Given two chromosomes

100010011110010010
010100010010000011

Choose a random bit along the length, say at position 9, and swap all the bits after that point

so the above become:

100010011010000011
010100010110010010

Mutation rate

This is the chance that a bit within a chromosome will be flipped (0 becomes 1, 1 becomes 0). This is usually a very low value for binary encoded genes, say 0.001

So whenever chromosomes are chosen from the population the algorithm first checks to see if crossover should be applied and then the algorithm iterates down the length of each chromosome mutating the bits if applicable.

A simple problem

Given the digits 0 through 9 and the operators +, -, * and /, find a sequence that will represent a given target number. The operators will be applied sequentially from left to right as you read.

Given the target number 23, the sequence 6+5*4/2+1 would be one possible solution.

If 75.5 is the chosen number then 5/2+9*7-5 would be a possible solution.

Encoding stage

Four bits are required to represent the range of characters used:

0:	0000
1:	0001
2:	0010
3:	0011
4:	0100
5:	0101
6:	0110
7:	0111
8:	1000
9:	1001
+:	1010
-:	1011
*:	1100
/:	1101

So now you can see that the solution mentioned above for 23, ' 6+5*4/2+1' would be represented by nine genes like so:

0110	1010	0101	1100	0100	1101	0010	1010	0001
6	+	5	*	4	/	2	+	1

Fitness function

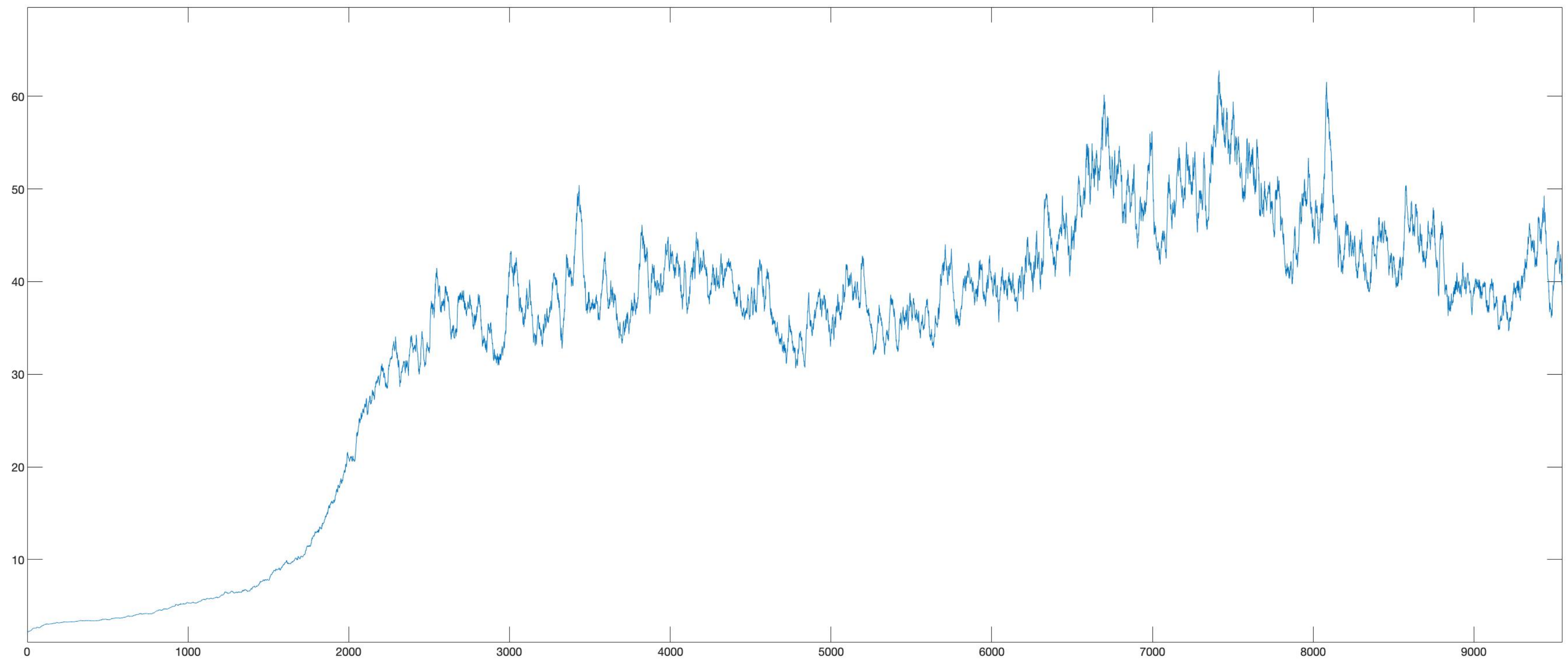
This can be the most difficult part of the algorithm to figure out. It really depends on what problem you are trying to solve but the general idea is to give a higher fitness score the closer a chromosome comes to solving the problem. A fitness score can be assigned that's inversely proportional to the difference between the solution and the value a decoded chromosome represents.

If we assume the target number for the remainder of the tutorial is 42, the chromosome mentioned above:

011010100101110001001101001010100001

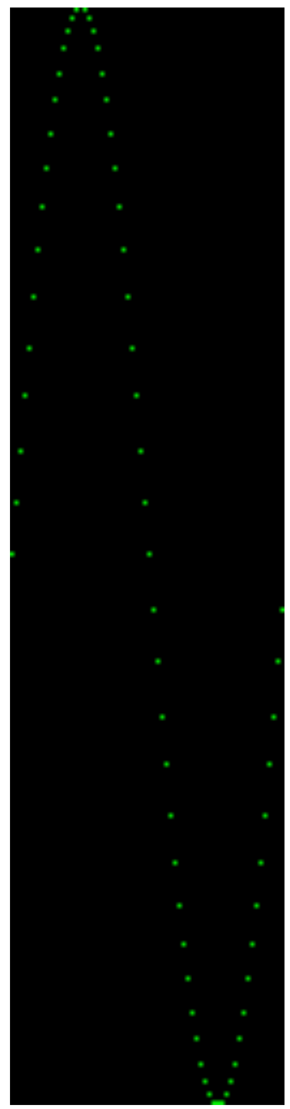
has a fitness score of 1/(42-23) or 1/19.

Example of fitness

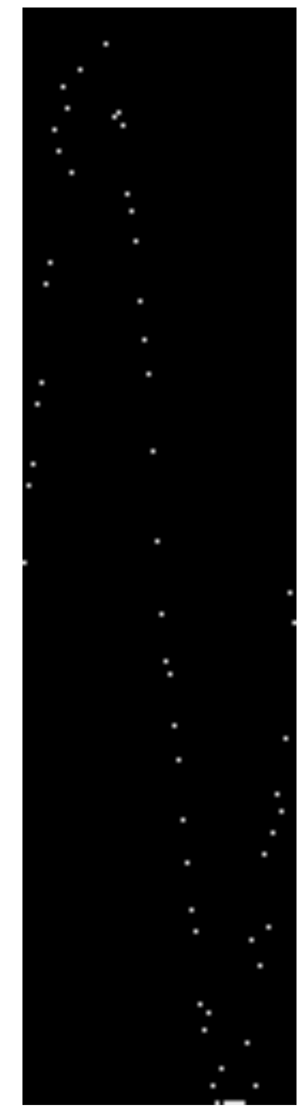


A more complex problem

Given a
Target function



How can we
Approximate it?



THANK YOU!

Suggested exercise:
try to implement a solution to the previous problem in Python!