A Short Introduction to Evolutionary Computation (in 50 minutes)

Fernando Lobo

University of Algarve

Outline

- Why EC in 50 minutes?
- The basic idea of a Genetic Algorithm
- Application example

Why EC in 50 minutes?

- If I only had 50 minutes to talk to someone about Evolutionary Computation, this is what I would show them.

- Two purposes:
  - to show you the basic idea quickly, without getting involved in too much details and theory.
  - to get you very motivated to study this topic.

Many variations on EC

- There's a variety of classes of Evolutionary Algorithms
  – Genetic Algorithms (GAs)
  – Evolution Strategies (ES)
  – Evolutionary Programming (EP)
  – Genetic Programming (GP)
  – …

- In this lecture, we will only look at the so-called Simple Genetic Algorithm.

What is a Genetic Algorithm (GA)?

- It is a search procedure based on the mechanics of natural selection and genetics.

- Often used to solve difficult optimization problems.

Evolution

- Evolution can be seen as an optimization process where living creatures constantly adapt to their environment.
  - the stronger survive and propagate their traits to future generations
  - the weaker die and their traits tend to disappear.

- Darwin's *survival of the fittest*.

How do GAs differ from other techniques?

- Work with a population of solutions, rather than a single solution.
- Use probabilistic rather than deterministic mechanisms.

Two essential components

- Selection
- Variation
  - recombination (or crossover)
  - mutation

When should we use a GA?

- When other simpler methods are not good enough.
- When we don't have much knowledge about the problem that we are trying to solve.
- … and when the search space is very large (making complete enumeration unfeasible).

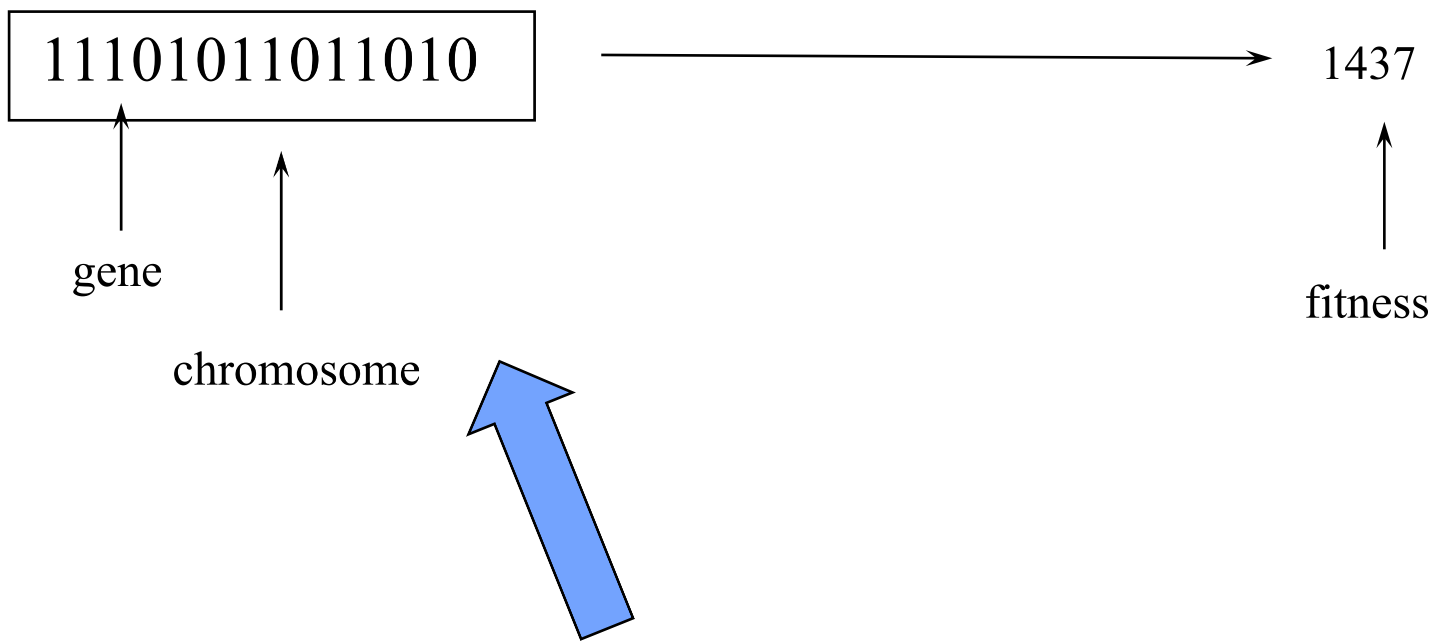
Encoding

- Problem solutions have to be encoded in some sort of structure (traditionally binary strings, but there’s many other possibilities).

- We need a way of quantifying the value (fitness) of any given structure (solution).
  - or given two solutions decide which is the better one.

- The fitness value is obtained through an objective function.

Terminology



This is the solution's DNA!

A Simple GA

Generate a population with *N* random individuals.
WHILE not happy with solution quality DO

1. Compute the fitness of every individual in the population.

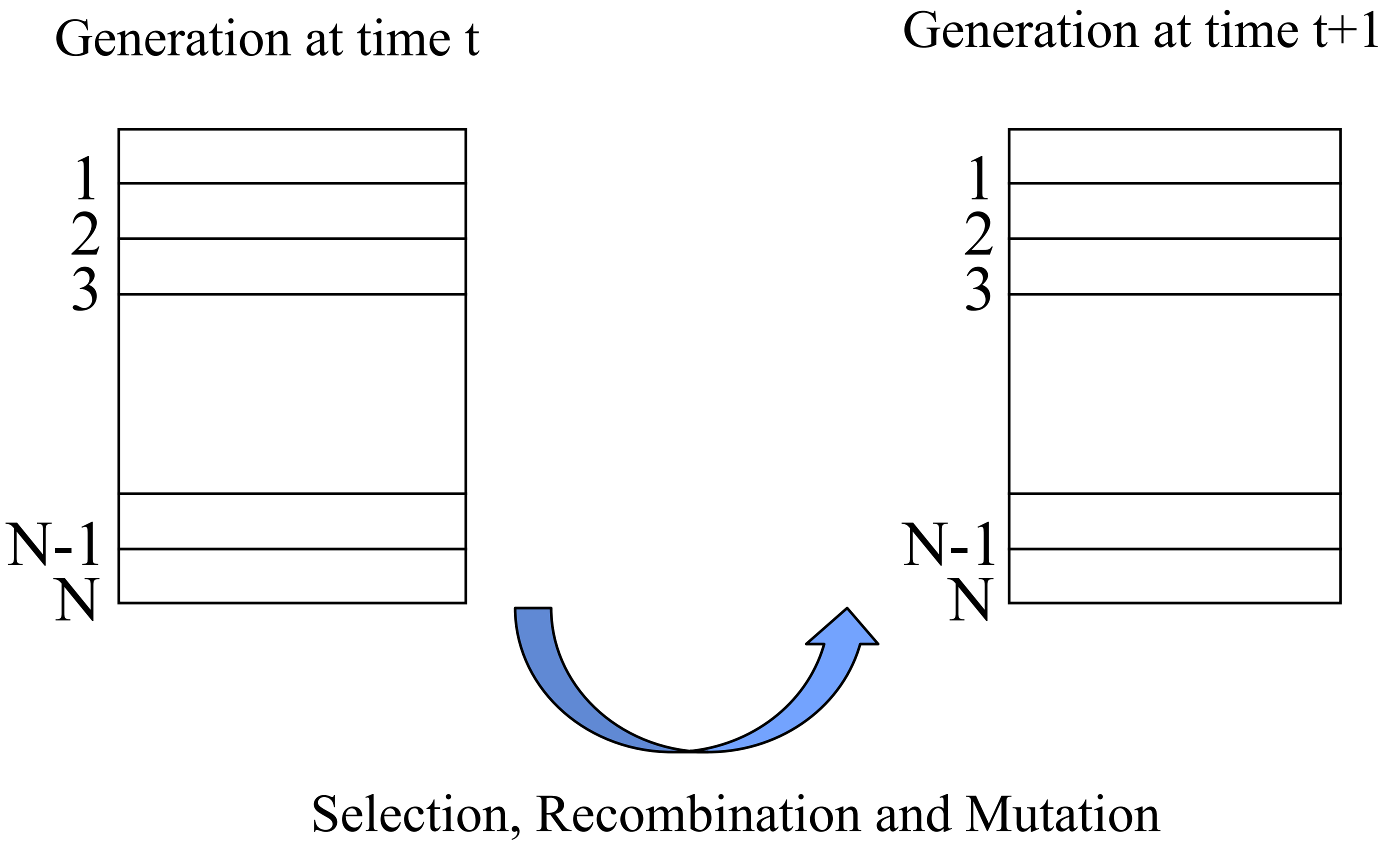
2. Select better individuals.

3. Do crossover between pairs of selected individuals with probability *Pc*.

4. Mutate each gene with probability *Pm*.

END WHILE

One iteration of the GA



Selection

- Simulates survival of the fittest.
- The stronger have a better opportunity to reproduce.
- The weaker tend to disappear from the population.
- Various methods to implement this operator (roulette wheel, ranking, tournament, and many others)

Example: binary tournament selection

- Pick a pair of individuals from the population.
- The winner survives, the loser dies.
- Repeat N times (N is the population size).

Example

before selection

individual	fitness
A	8
B	2
C	4
D	5

after selection

individual	fitness
A	8
C	4
A	8
D	5

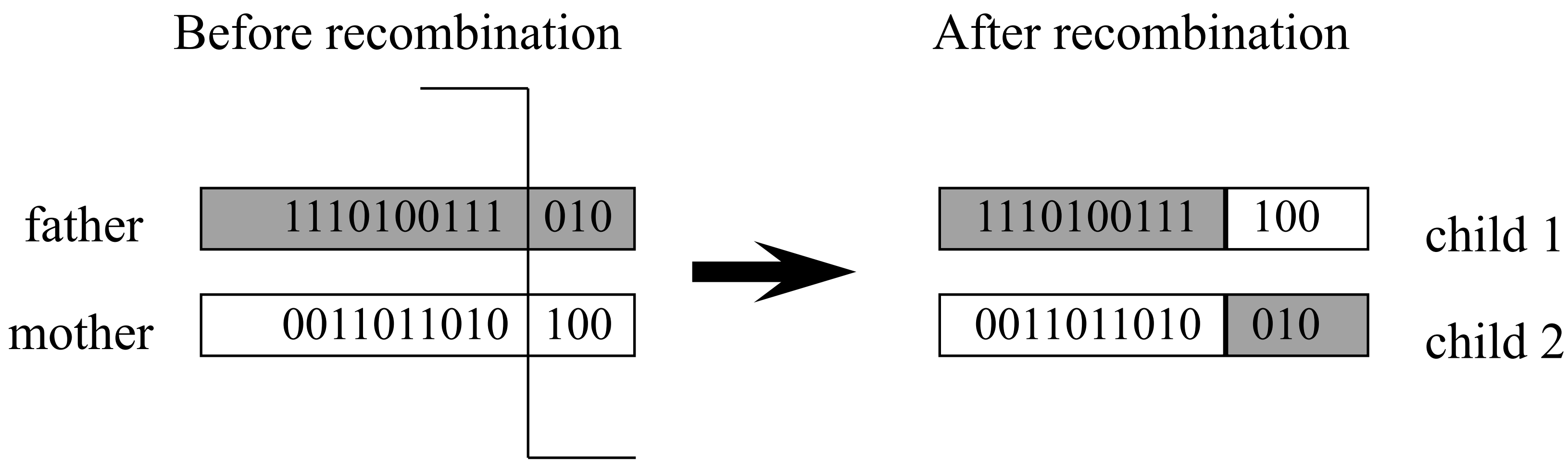
4 tournaments: (A,D) (B,C) (A,B) (C,D)

4 winners: A C A D

Crossover

- Recombine 2 individuals (father and mother) to obtain two new individuals (the children).
- In the previous example, A recombines with C, and A recombines with D.

Example of crossover



Mutation

- With probability *Pm*, flip a gene from 0 to 1, or from 1 to 0.

- In traditional GAs, this operator is typically used with a low probability.

before mutation	after mutation
1100101111	1100101101

gene 9 was mutated

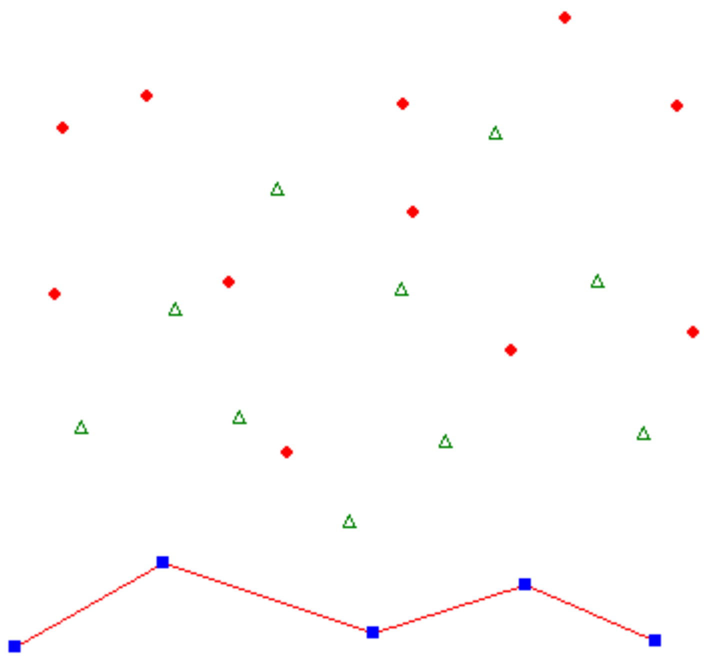
Why do they work?
(a very informal explanation)

- We can make an analogy with the way humans are creative and innovative (Goldberg).

- Humans are creative and innovative when they combine notions that work well in one context, with notions that work well in some other context.

- Likewise, GAs can be creative when combining pieces of a good solution, with pieces of another good solution.

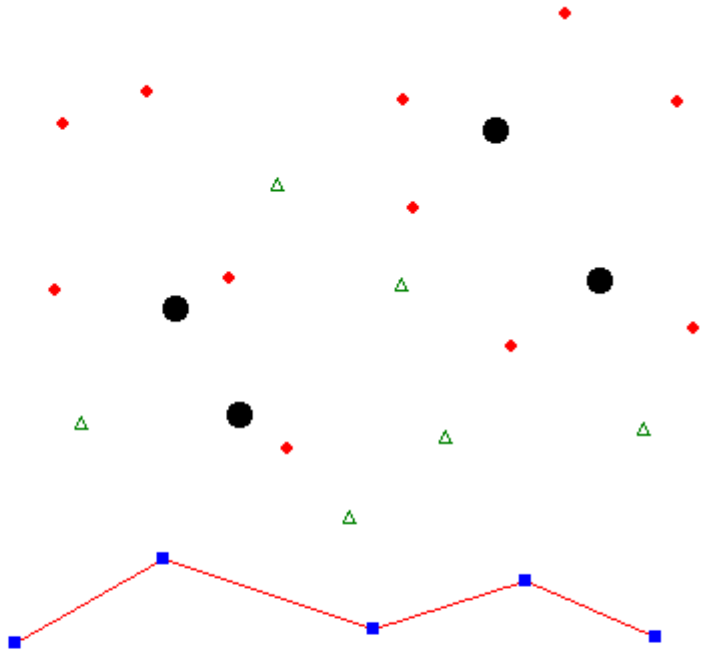
Example: a network expansion problem



10 decision variables (10 triangles).

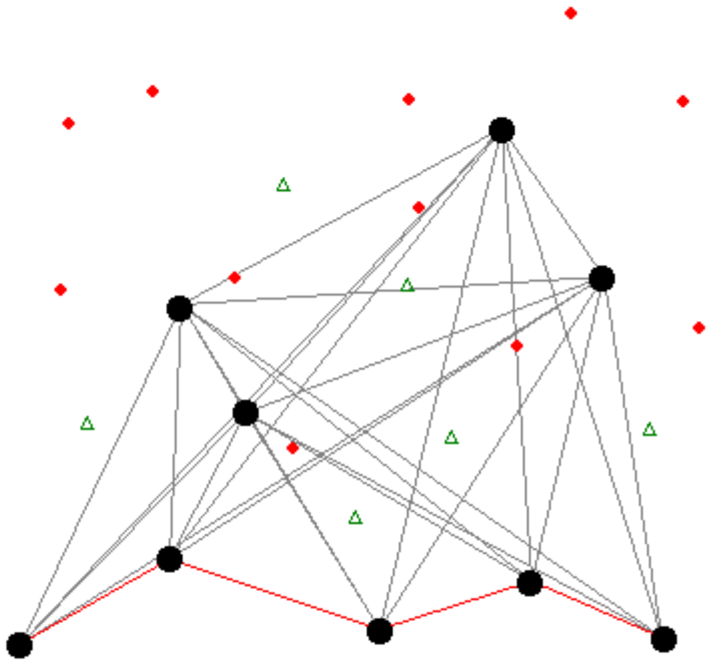
Want to expand the network to deliver electricity to the houses (red dots). The houses can connect directly to existing transformers (blue squares), or they can connect to a transformer (green triangles). The goal is to expand the network using the least amount of money.

Objective function in 3 steps

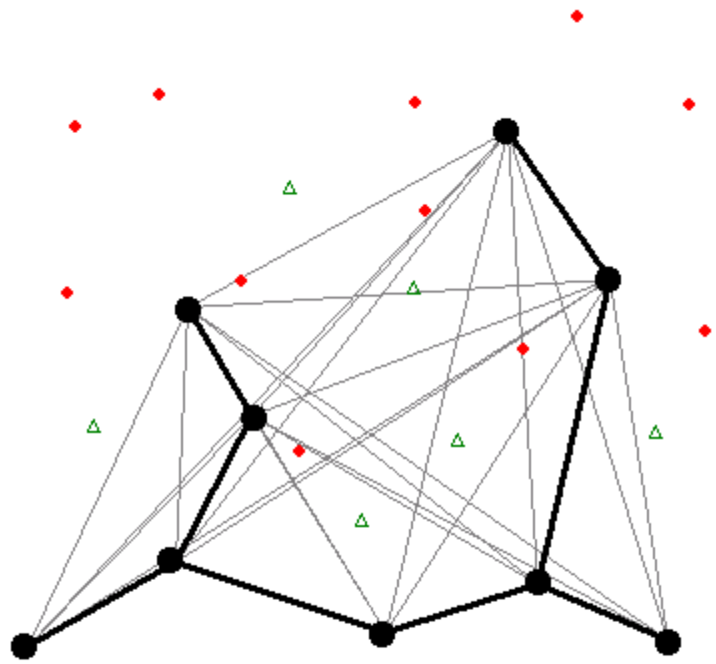


Solution: 0110000110

step #1

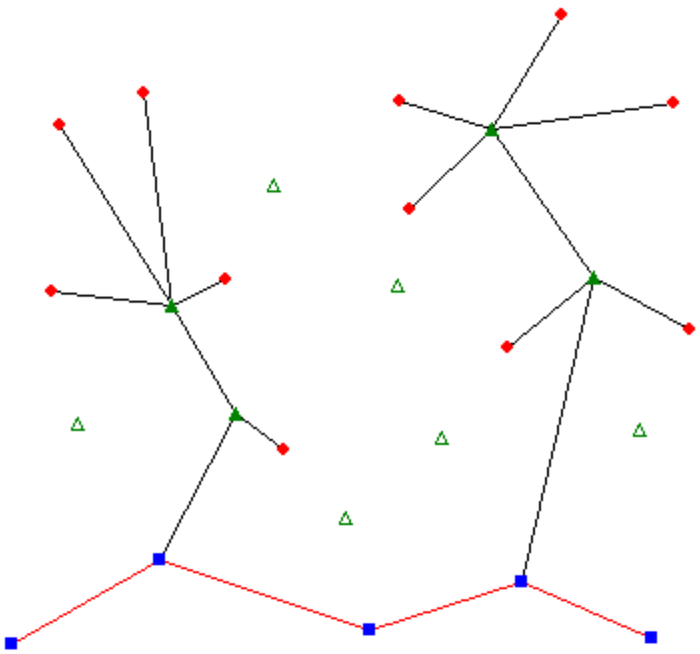


step #2



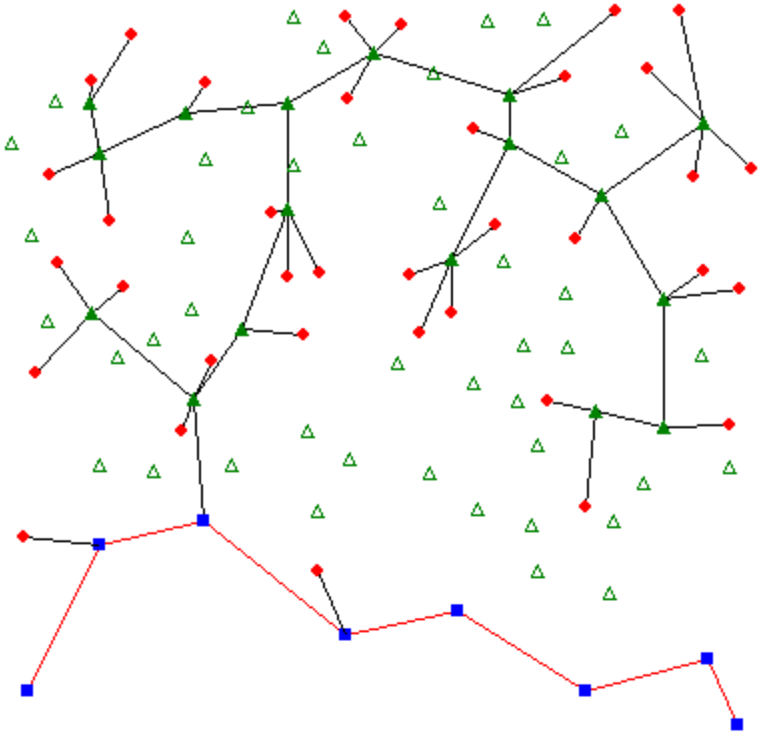
Minimum Spanning Tree

step #3



Network corresponding to solution 0110000110

GA discovered this solution after a few number of generations



Example with 60 decision variables.

$2^{60} \approx 1000000000000000000$ possible networks!

Looking ahead…

- We have just seen the very basics of GAs
- There's many variations on it, and there's other classes of related algorithms.
- We have the rest of the course to learn about them…

DEMO

- Let’s look at the network expansion demo now.