

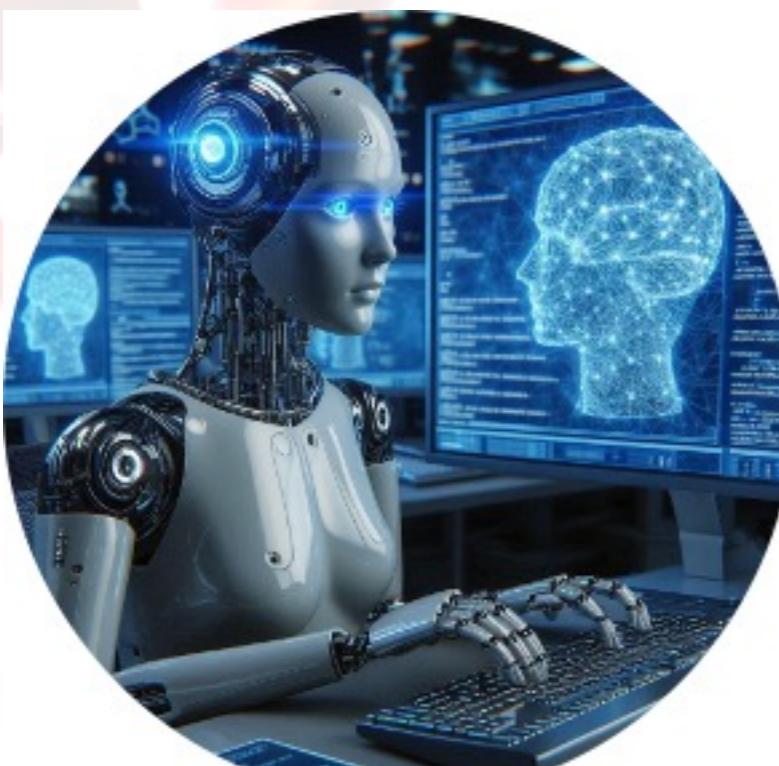
Heuristics & Metaheuristics for Optimization & Learning



Mario Pavone

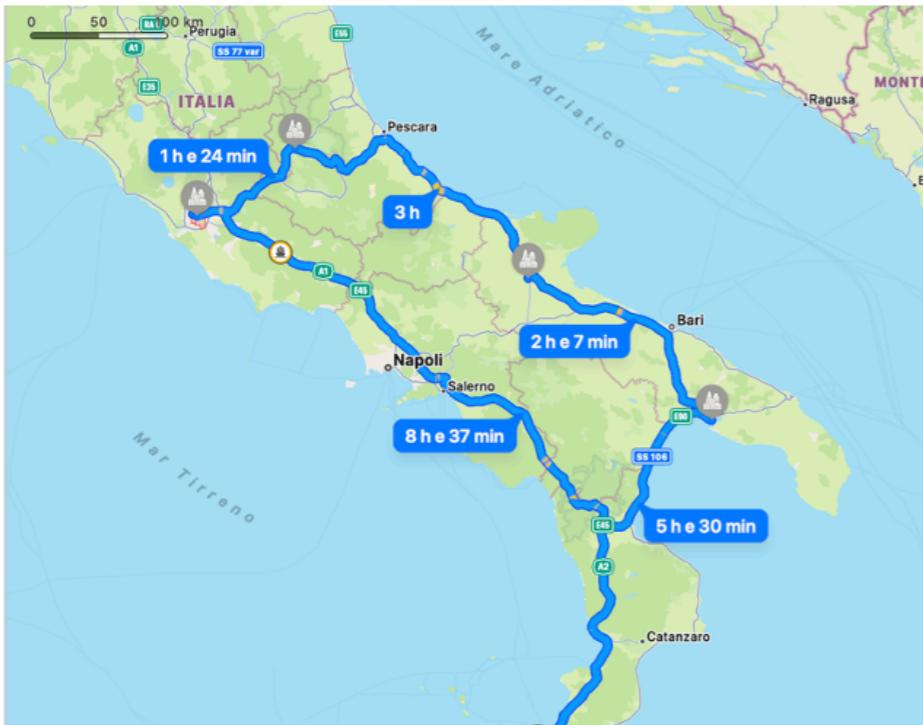
mpavone@dmi.unict.it

<http://www.dmi.unict.it/mpavone/>

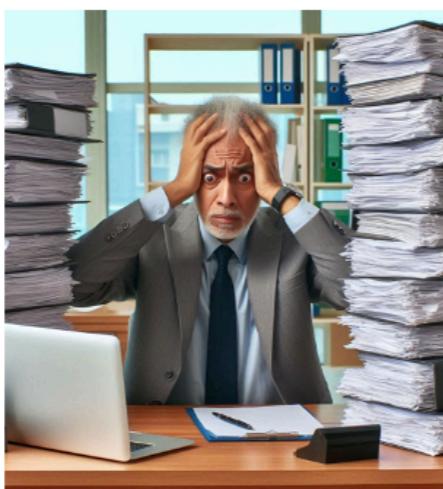


Why a problem is Hard?

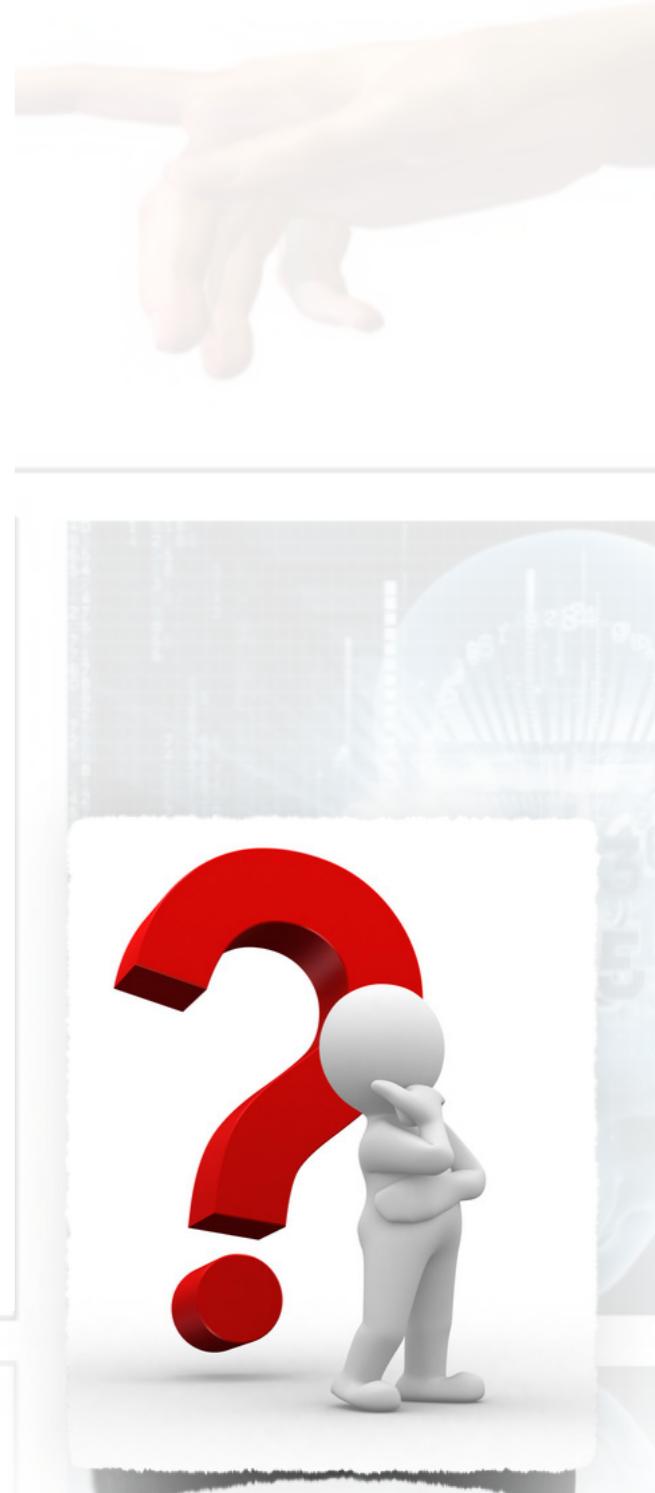
Time



Uncertainty



Big Data





Look the nature and
be inspired!

Population-Based Algorithms

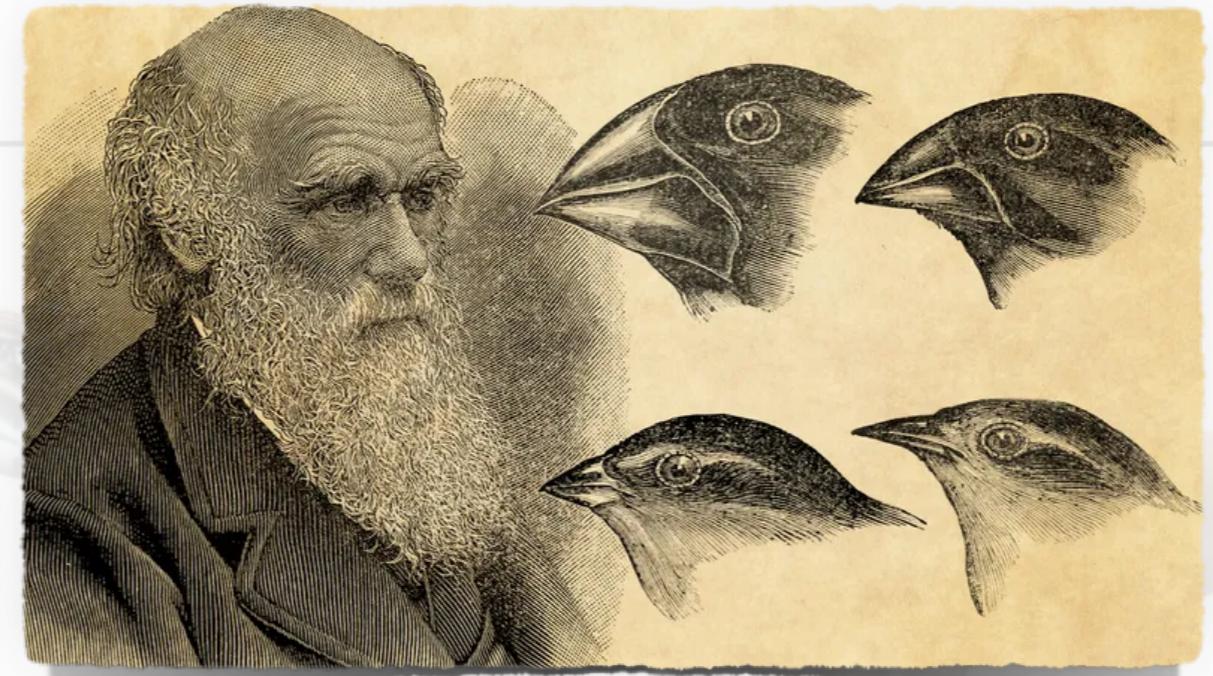
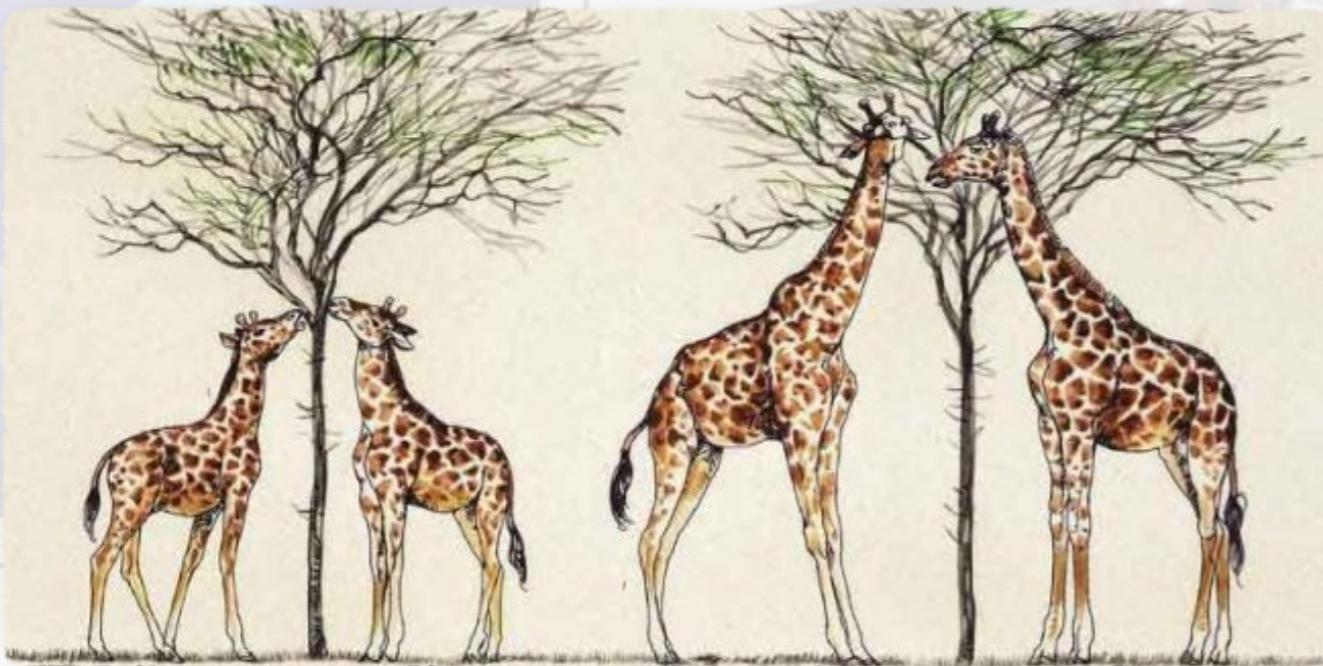
Look the nature and
be inspired!



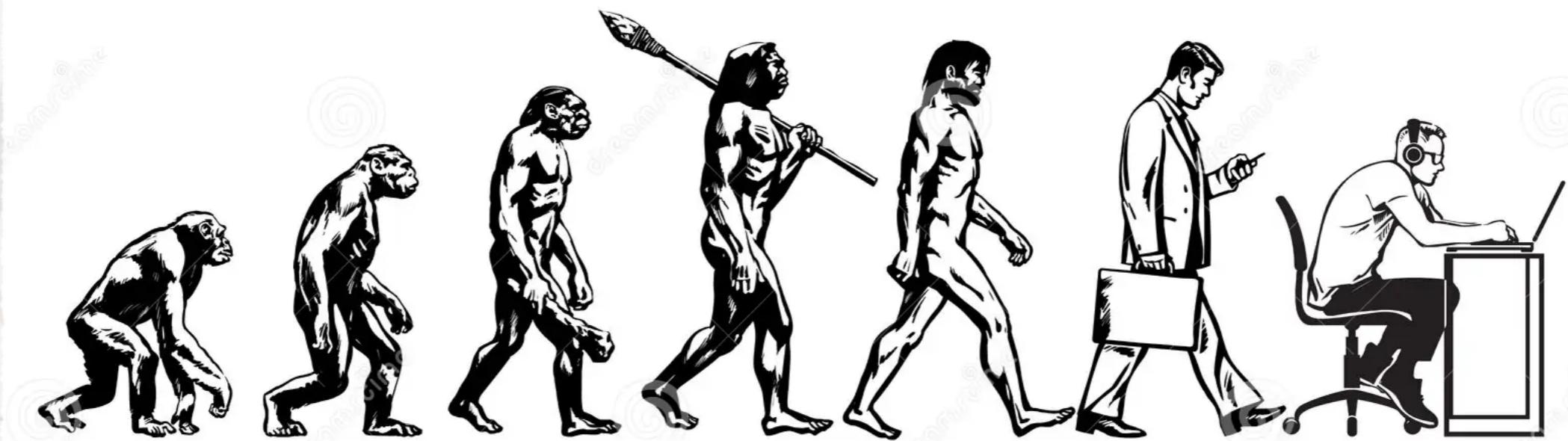


HOW

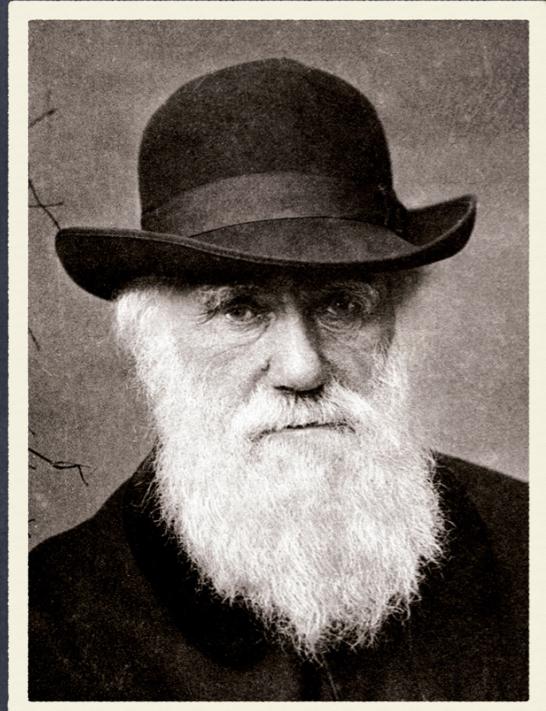
A large red 3D text "HOW" is shown. A magnifying glass is positioned over the letter "O", focusing on it. The background is a light blue gradient.



Teoria dell'Evoluzione



Teoria dell'Evoluzione



Charles Darwin

Selezione Naturale del *migliore*

Crossover

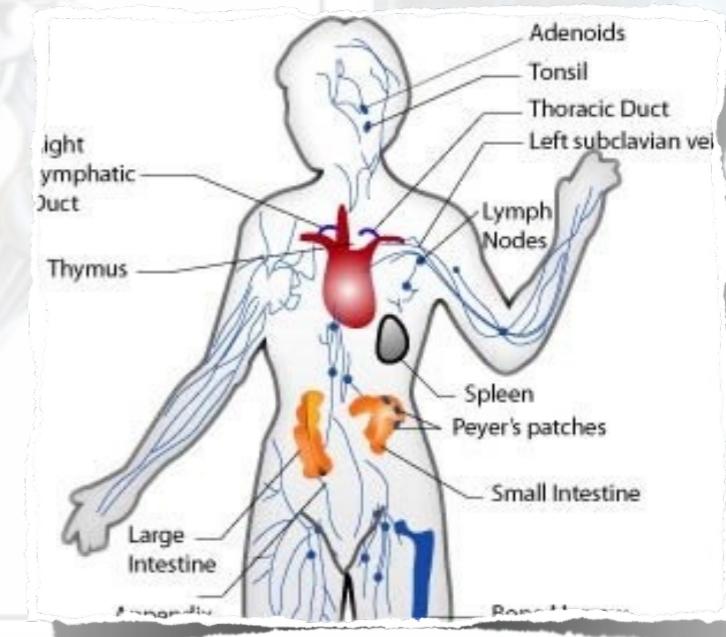
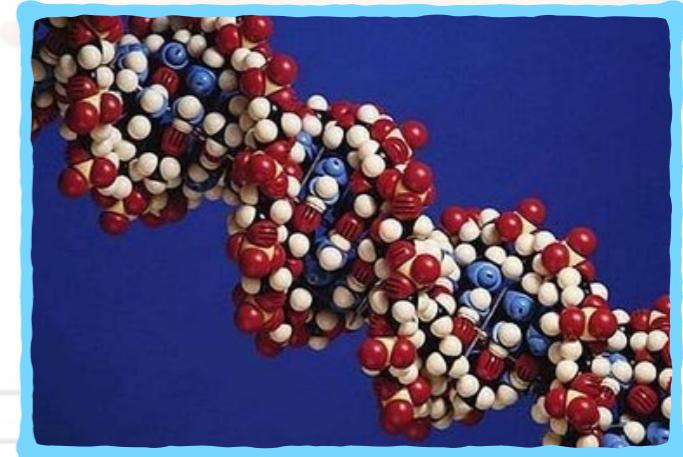
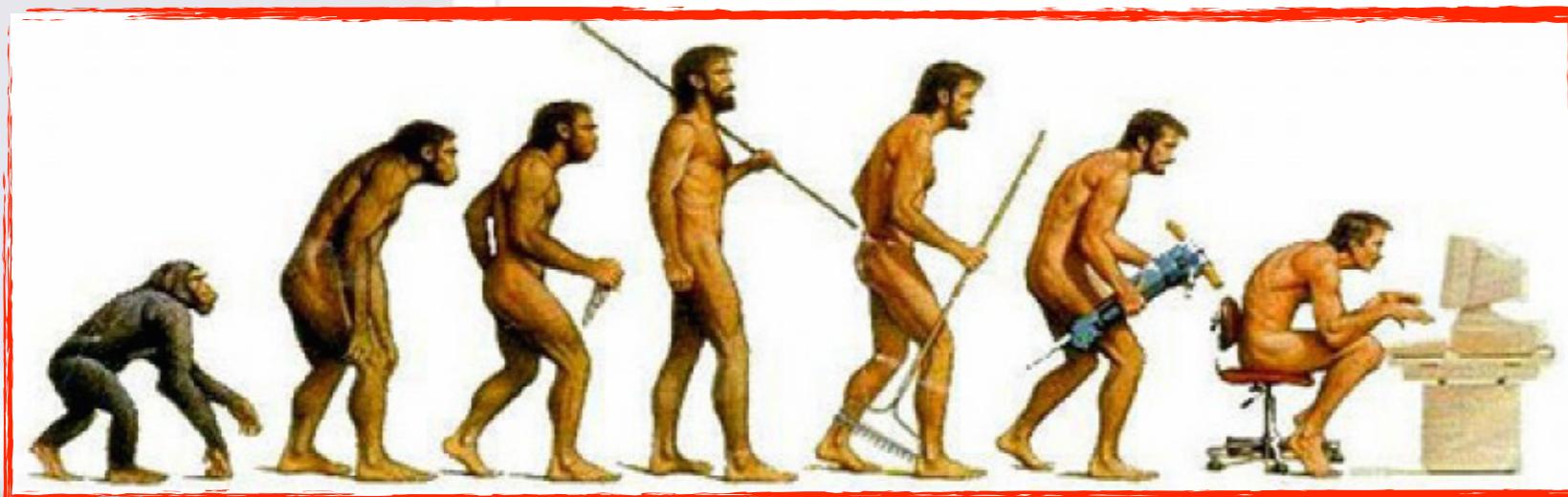
Adattamento

Evoluzione

Mutazione

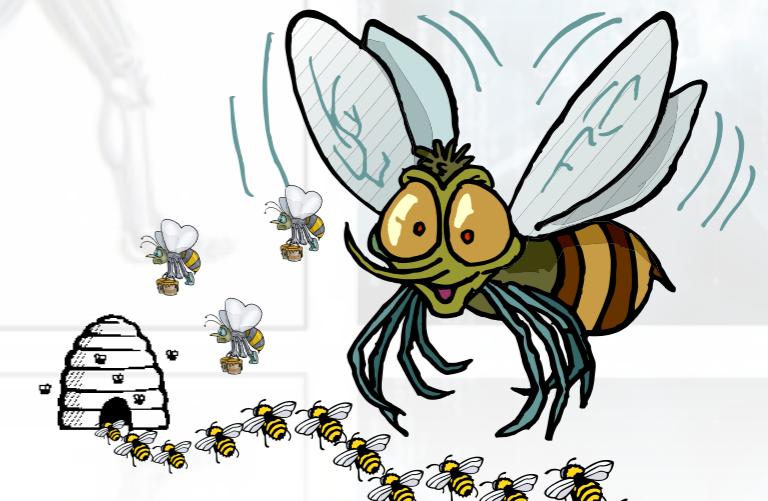
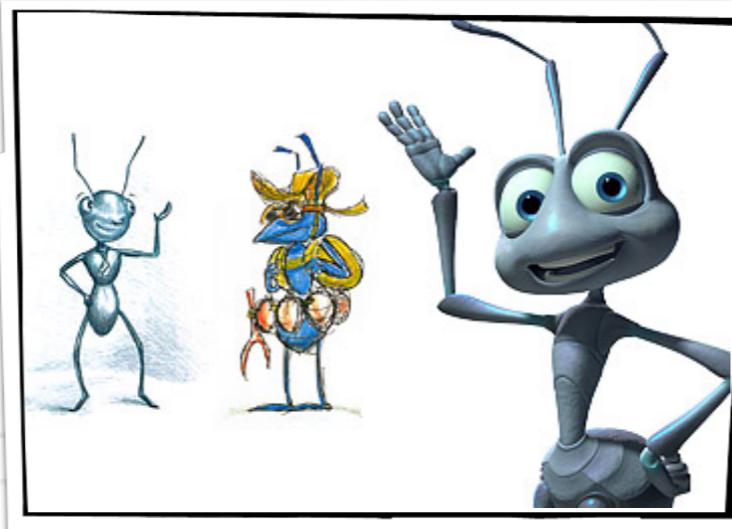


Algoritmi Bio- & Nature-Inspired

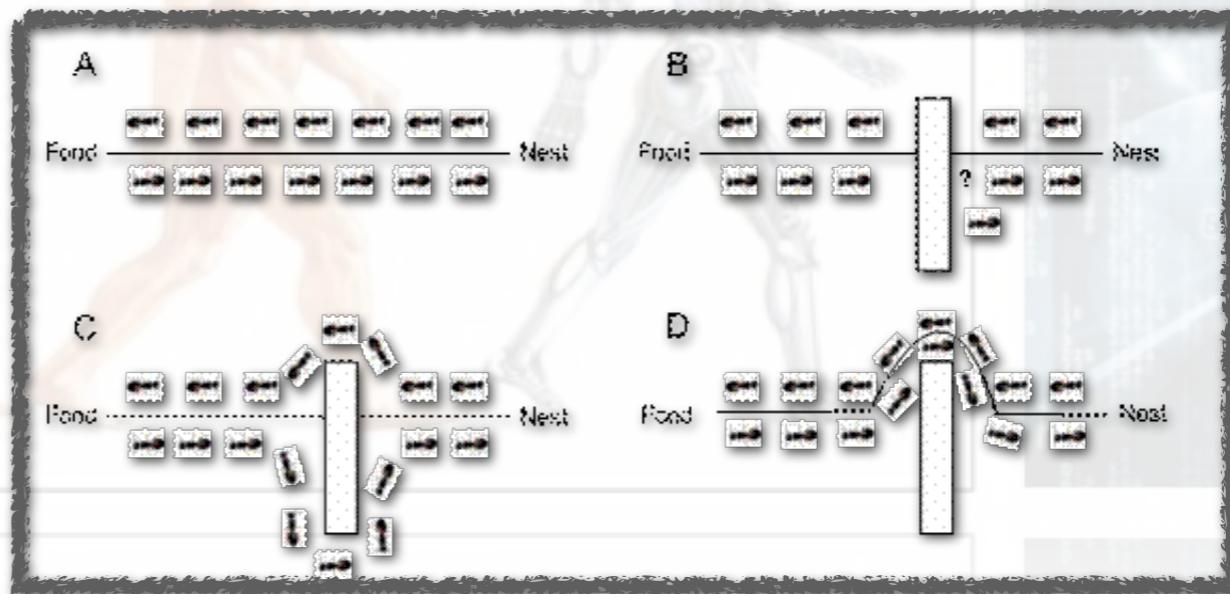
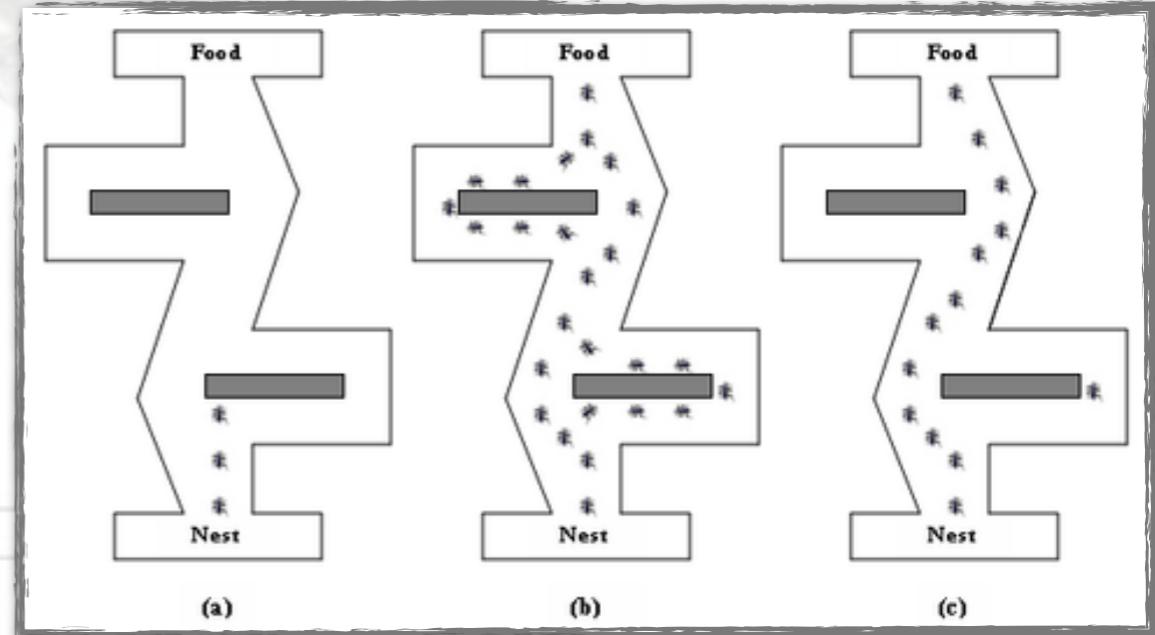


Swarm Intelligence

Comportamento Collettivo, Decentralizzato & Self-Adaptive



Colonne di Formiche



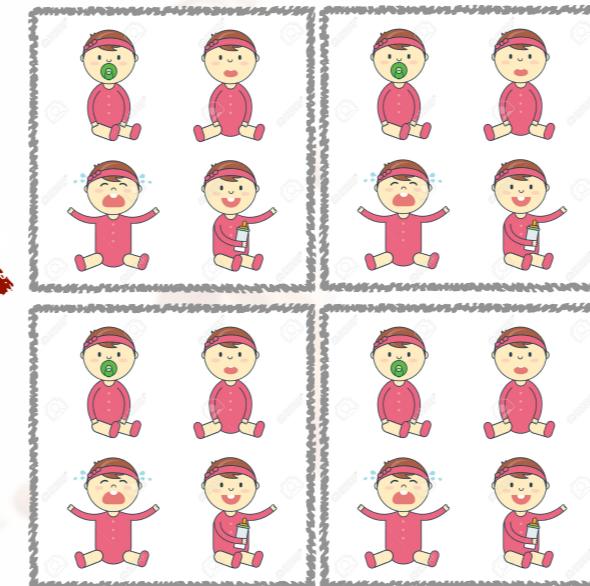
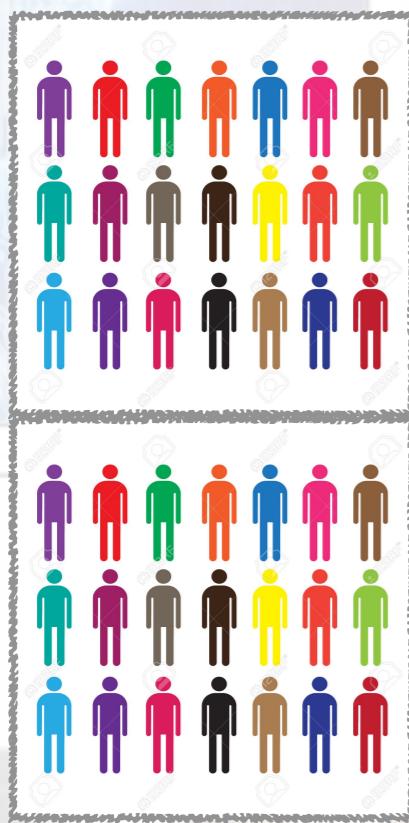


Come

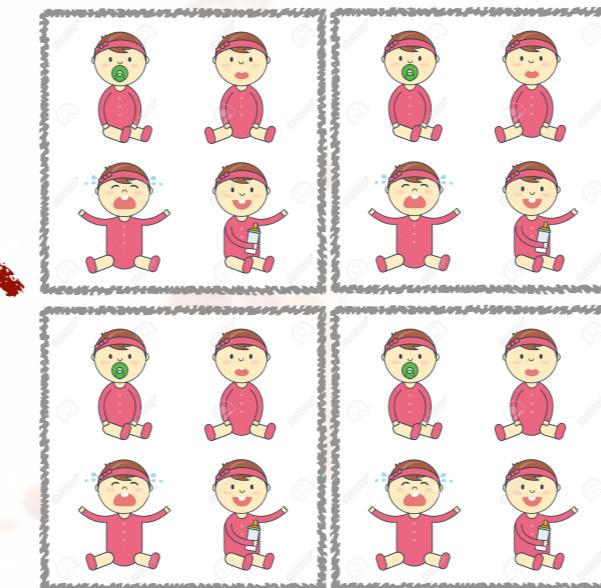
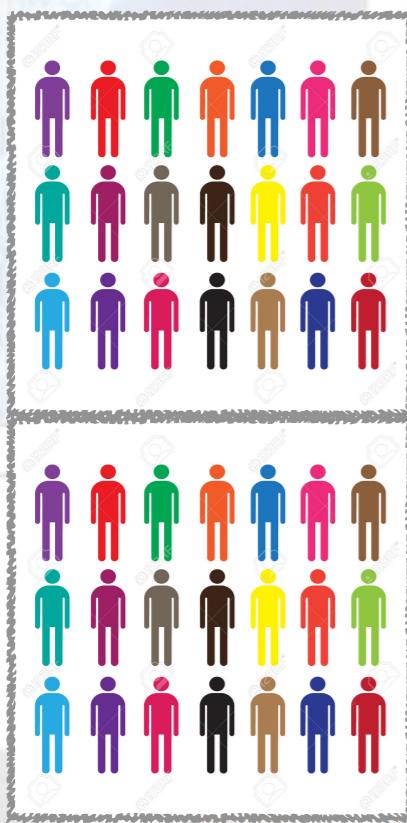
funzionano?



Evoluzione Naturale

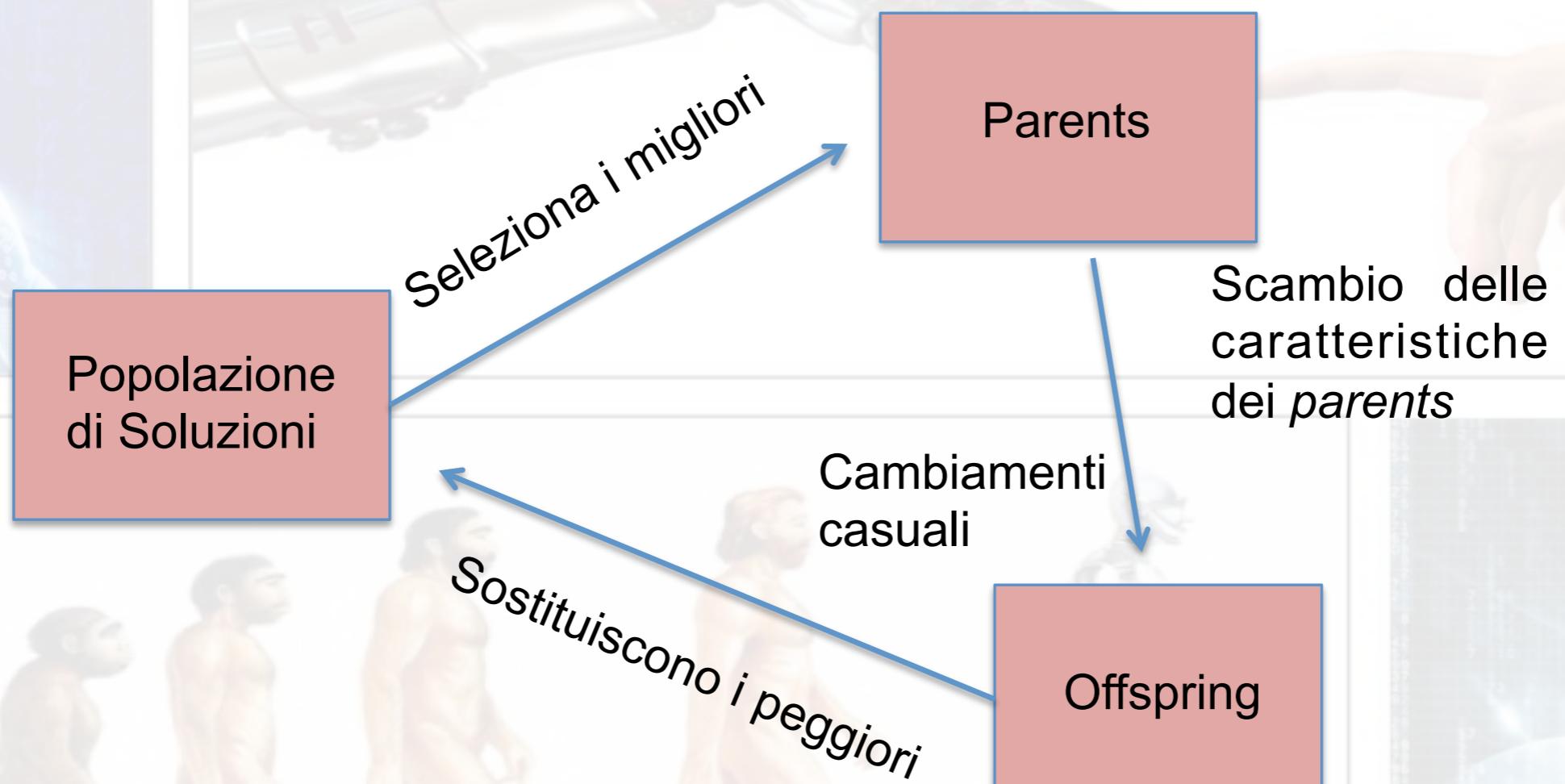


Evoluzione Naturale

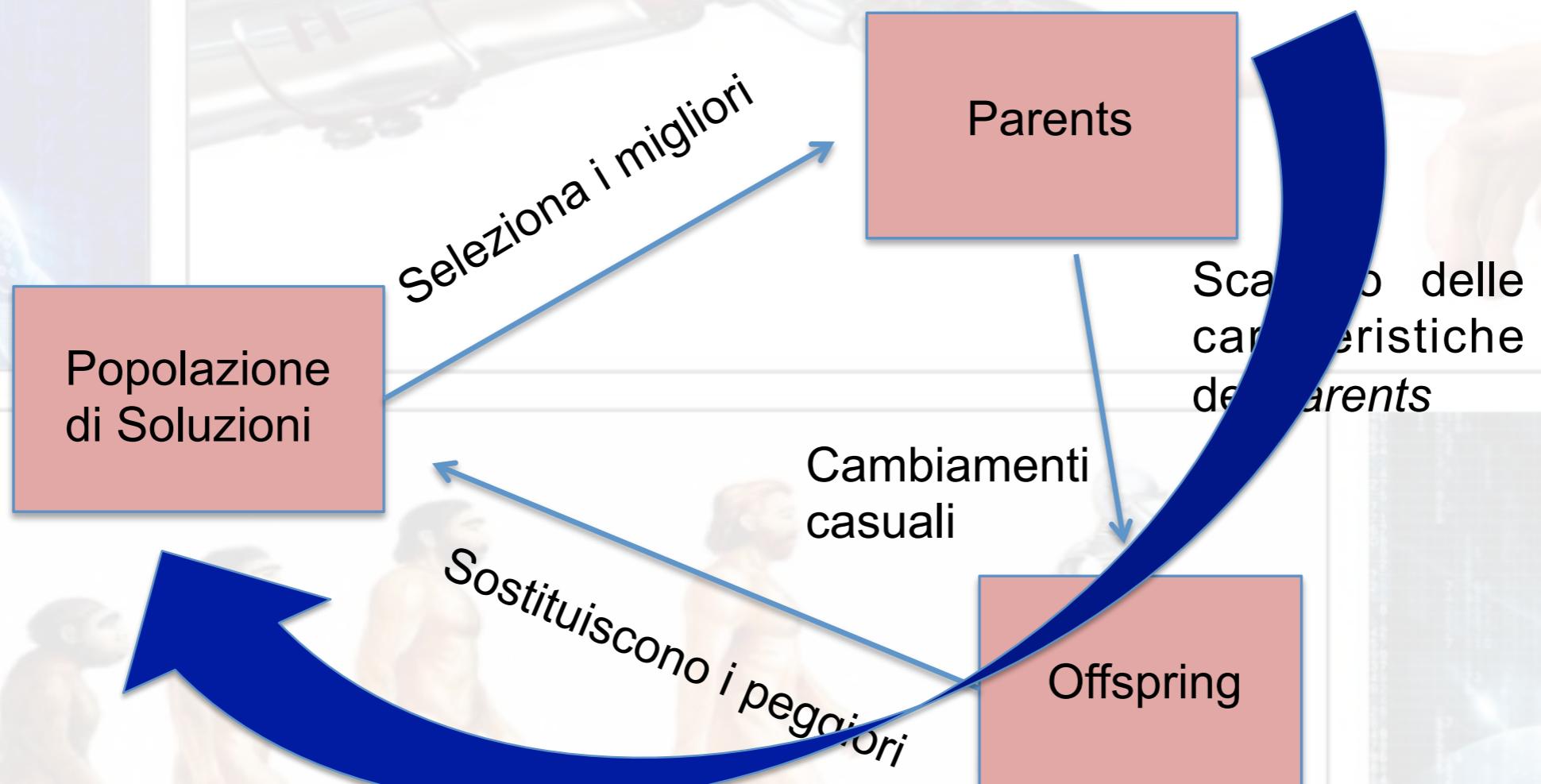


LOOP

Evoluzione Artificiale



Evoluzione Artificiale



A Simple Scheme of GA

1. Generate the initial **population** $P(0)$ at random, and set
 $i \leftarrow 0$;
2. REPEAT
 - (a) Evaluate the fitness of each individual in $P(i)$;
 - (b) **Select** parents from $P(i)$ based on their fitness in $P(i)$;
 - (c) **Generate** offspring from the parents using *crossover* and *mutation* to form $P(i + 1)$;
 - (d) $i \leftarrow i + 1$;
3. UNTIL halting criteria are satisfied

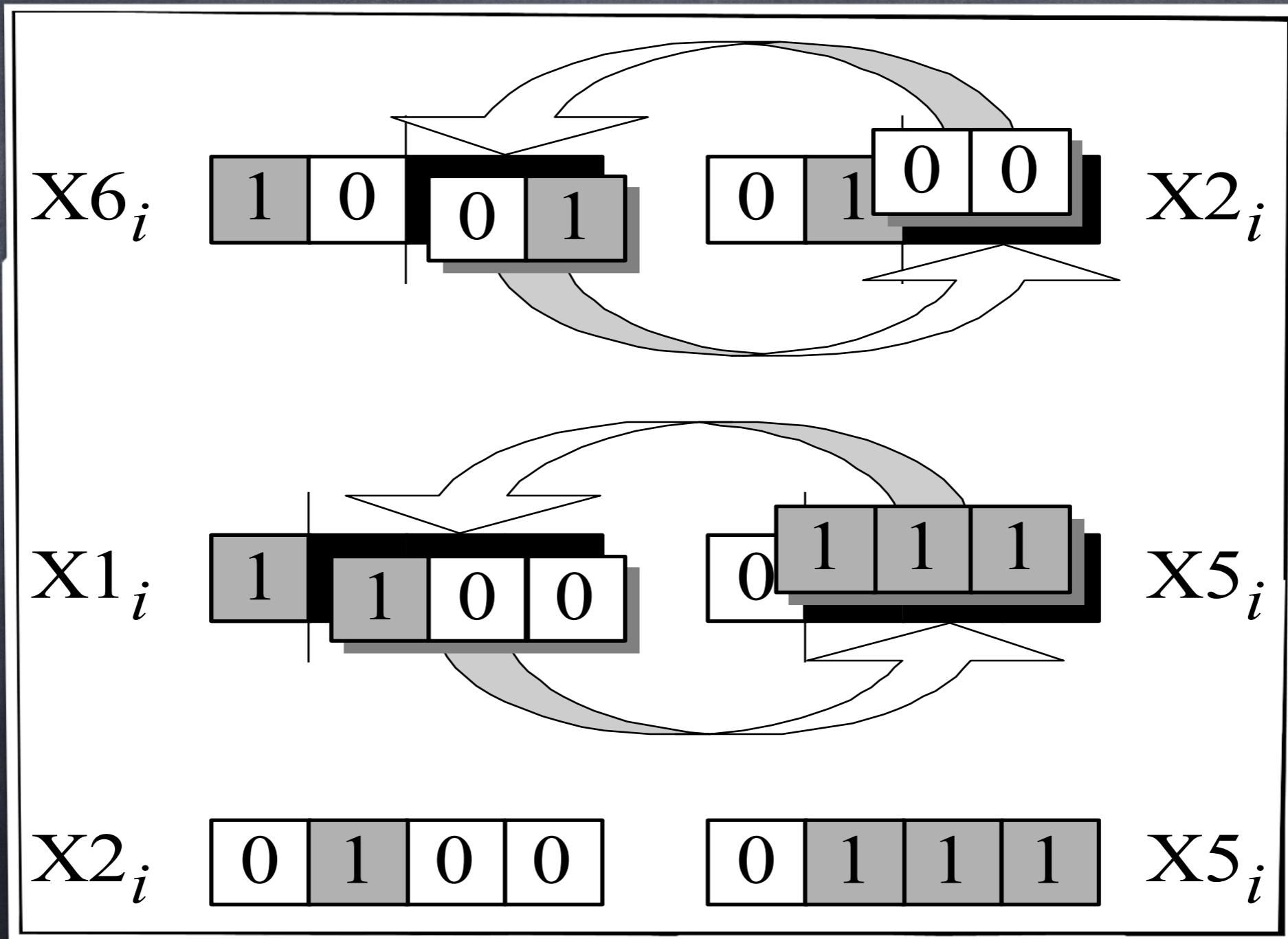
A Genetic Algorithm: how it works

- Randomly generate a set of possible solutions to a problem, representing each as a fixed length character string
- Test each possible solution against the problem using a fitness function to evaluate each solution
- Keep the best solutions, and use them to generate new possible solutions
- Repeat the previous two steps until either an acceptable solution is found, or until the algorithm has iterated through a given number of cycles (generations)

Basic Operators of GA

- **Reproduction:** It is usually the first operator applied on population. Chromosomes are selected from the population of parents to cross over and produce offspring. It is based on Darwin's evolution theory of "Survival of the fittest". Therefore, this operator is also known as '**Selection Operator**'.
- **Cross Over:** After reproduction phase, population is enriched with better individuals. It makes clones of good strings but does not create new ones. Cross over operator is applied to the mating pool with a hope that it would create better strings.
- **Mutation:** After cross over, the strings are subjected to mutation. Mutation of a bit involves flipping it, changing 0 to 1 and vice-versa.

CrossOver: an example



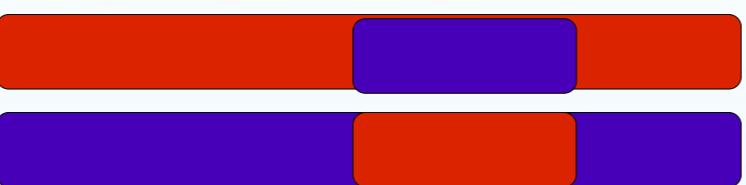
Kinds of CrossOver

- **Crossover**

- Single Point



- Two point



- Uniform

