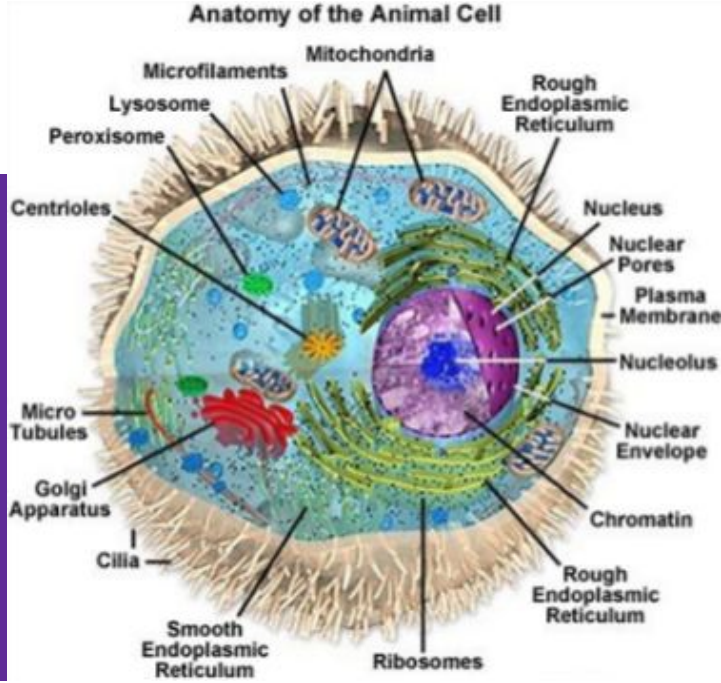
Genetic Algorithms

IN104
Natalia Díaz Rodríguez, ENSTA
Institut Polytechnique Paris



Problem and Approach

- **Optimization**: a branch of mathematics that searches and seeks solving analytically or numerically problems that consist on determining the best element of a set, given some qualitative criteria.
- **Difficulty**: often computing solutions for NP-complete problems
- **Solutions**:
  - Steepest Descent
  - Simulated Annealing (Monte Carlo)
  - ***Genetic Algorithms***

Genetic Algorithms (GA)

- Problem resolution techniques that require optimization
- A **metaheuristic** belonging to the larger class of Evolutionary Algorithms (EA)
- Based on the *Evolution Theory of Darwin*
- Inspired by a *natural selection* metaphor:
    - Keep best N hypotheses at each step (**selection**) based on a fitness function.
    - Have a pairwise **crossover** operator, with optional **mutation** to give variety

Natural Selection

Principles:

- Preservation of the favorable variations
- Rejection of unfavorable variations

Individuals that have an advantage have more probabilities of surviving:

“Survival of the fittest” => **Fitness Function**

=> One of the trickiest part to design!

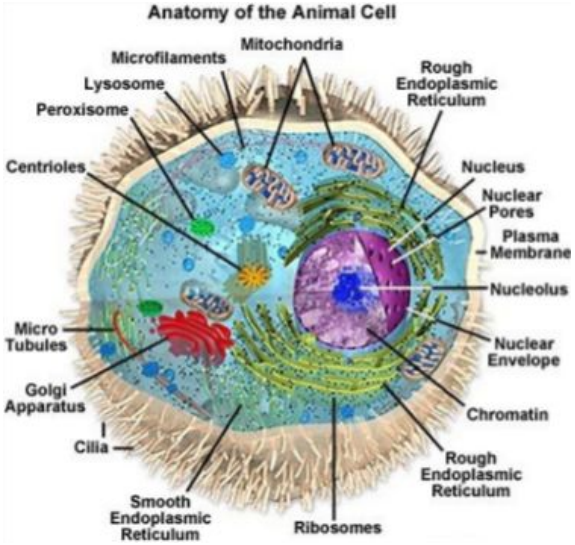
History

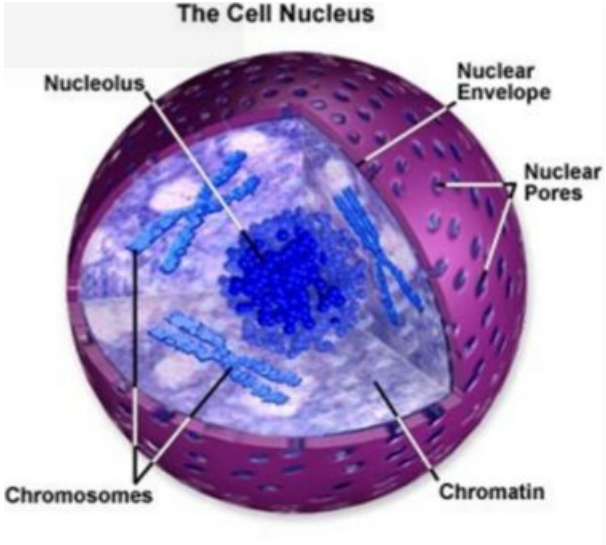
- **EA**: Developed in the 60s
- **GA**: Created by John Holland in the 70s.

The intuition

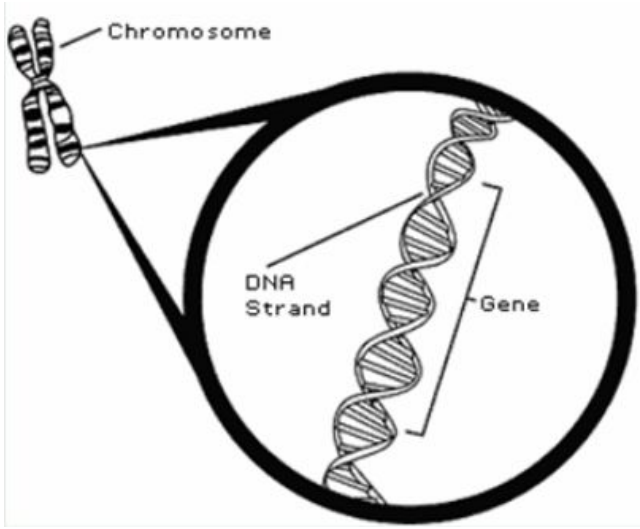
- Based on the model of life beings who are formed by **cells** that contain a **nucleus**
- The genetic information is stored in the **chromosome** that is encoded in the DNA

The cell





Detail of chromosome



Model Representation

Representation:

- 010101110010001010010101 <= Genetic information

Population:

- 100010110100101010101001
- 010101010101010100101001
- 010101011111100000101001
- 000001010101010100101001
- 010101010101010111111101 <= Gene pool

GA Phases and Termination

- The population evolves following certain rules
- Until:
  - an individual that corresponds to a criterion is produced
  - The population has been evolved for X iterations
- **Iteration**: Reproducing genetic information. Each one involves:
  - **Mitosis**: Duplication (recopying existing information), reproduction
  - **Meiosis:** Division

Genetic Algorithms Operators

GA are commonly used to generate high-quality solutions to **optimization and search** problems by relying on bio-inspired **operators:**

- **Selection**: filtering out the less fit
- **Cross-over**: recombination (‘*mating’*)
- **Mutation**: part of the initial information is lost and replaced with other information

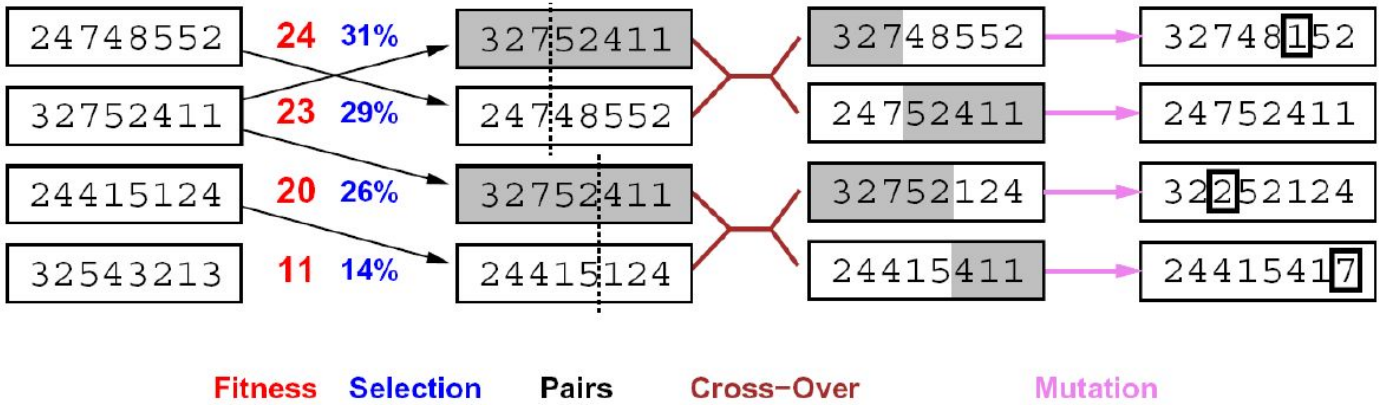
Randomness

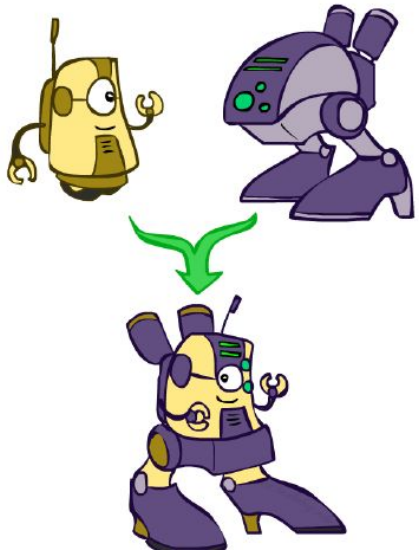
- During reproduction, **errors** are produced.

Thanks to these, the population varies to produce a better element, avoiding **local minima**.

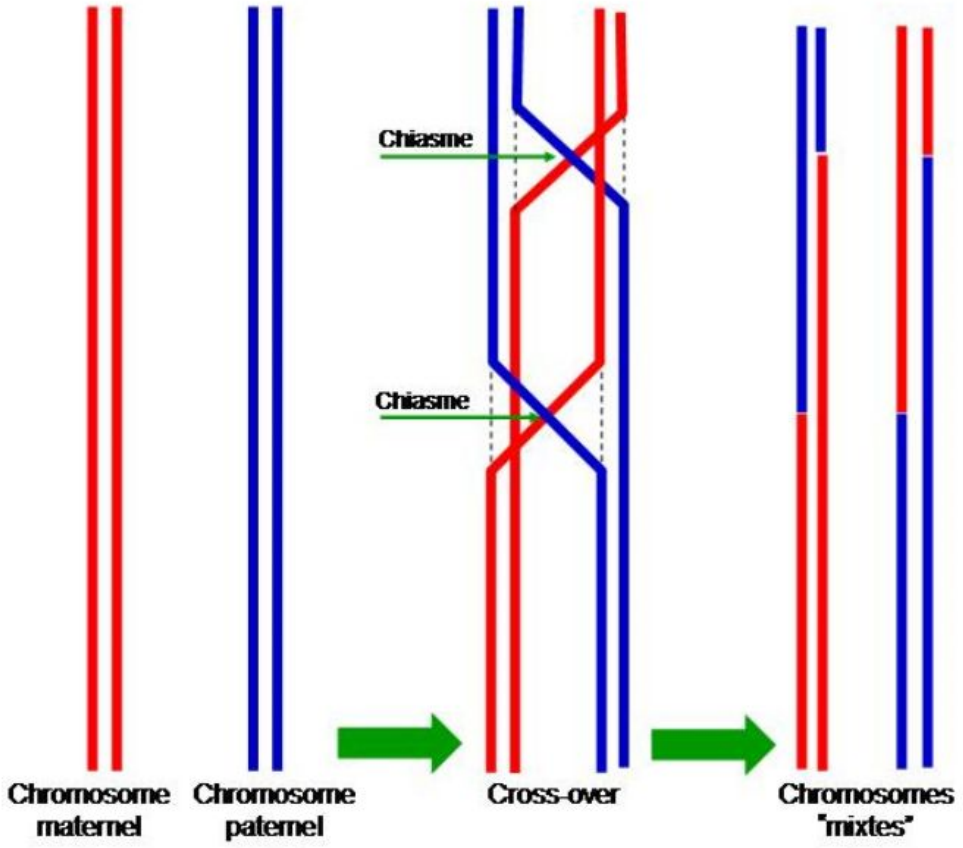
=> Mutation

GA Bio-inspired operators:

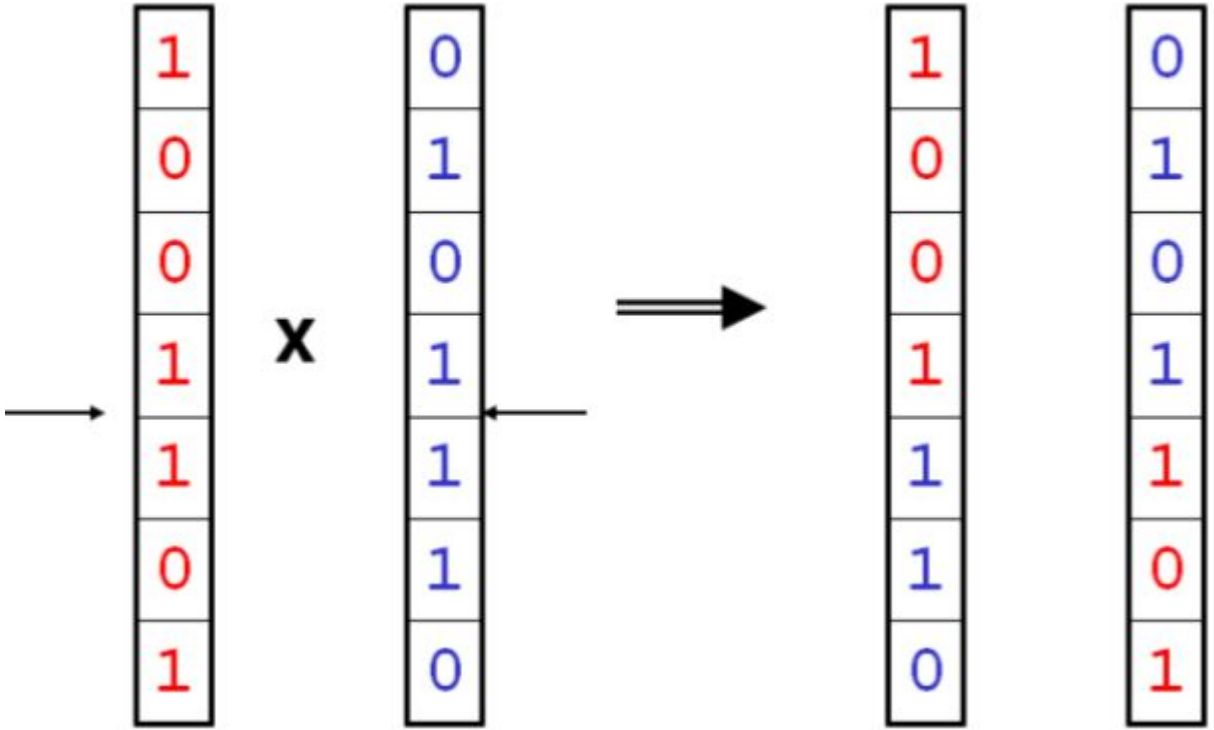




GA Bio-inspired operators: Cross-over

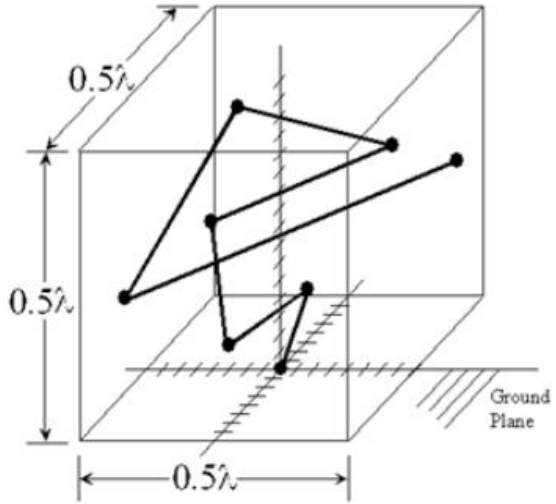


GA Bio-inspired operators: Cross-over



Genetic Algorithms Applications

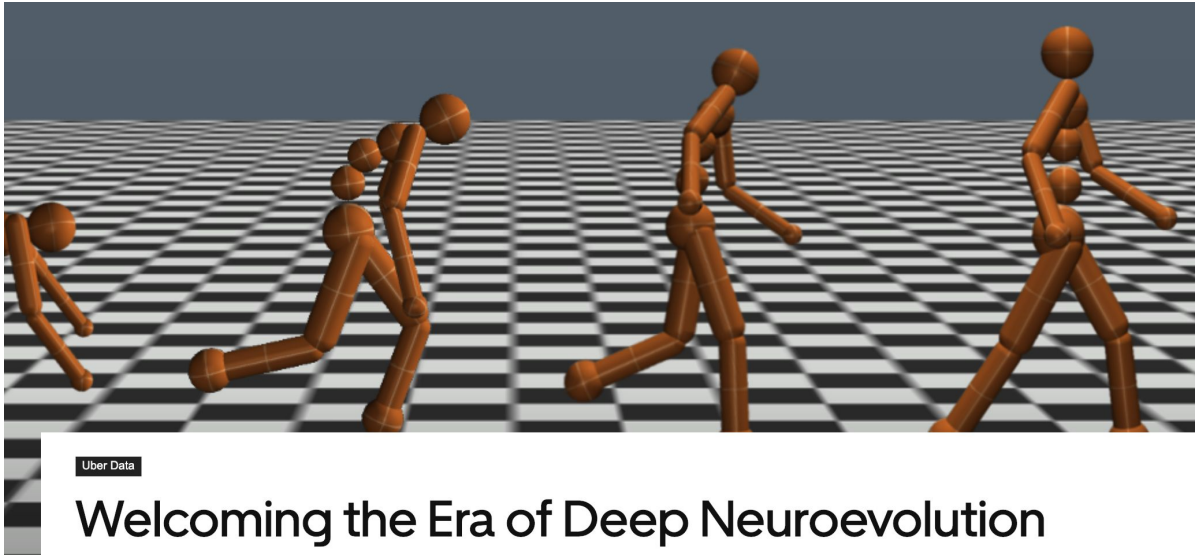
- Finance
  - Stock market: data mining, optimisation
- Antennas Modelling



- Networks
  - Network intrusion detection

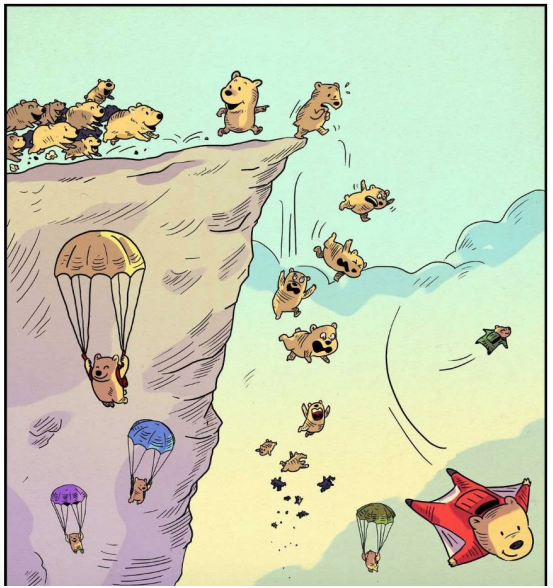
Genetic Algorithms Applications

- Learning locomotion skills, robotics, Atari games…



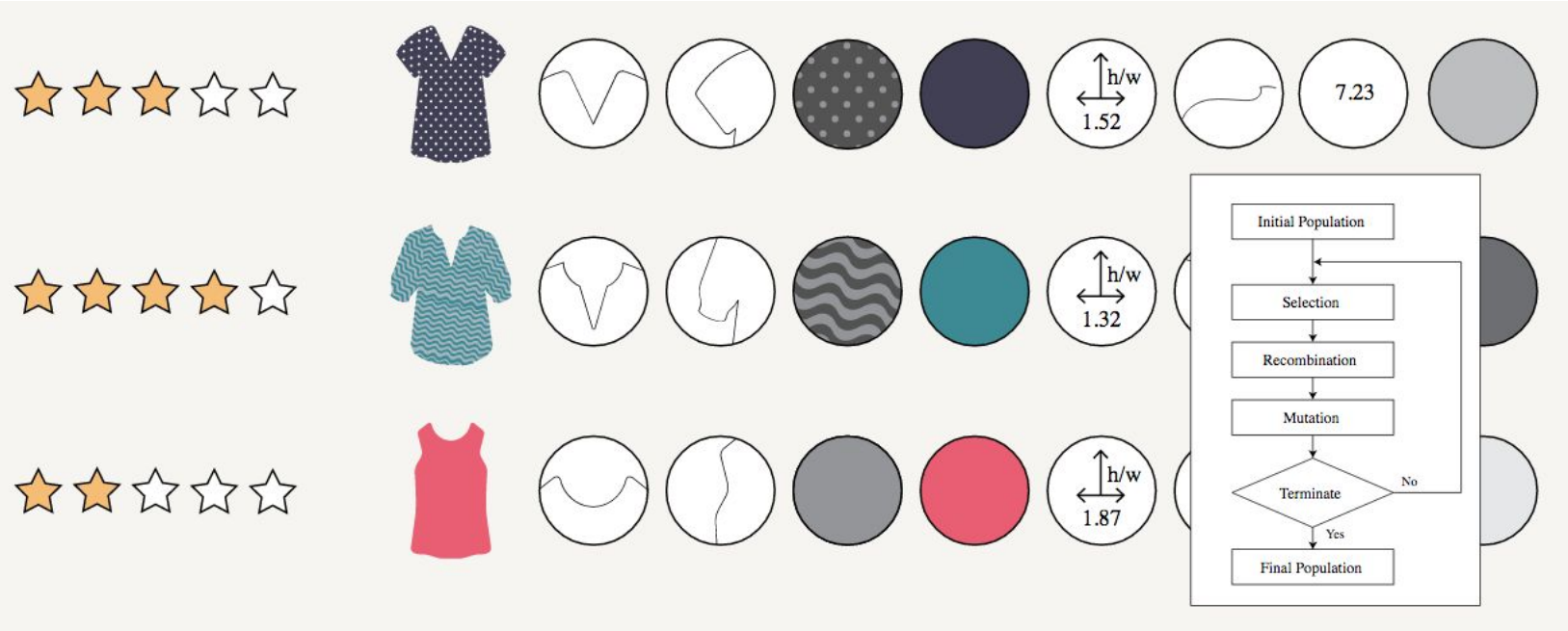
A Visual Guide to Evolution Strategies

OCTOBER 29, 2017



Genetic Algorithms Applications

- Learning locomotion skills, robotics, Atari…
- Fashion design



Acknowledgements

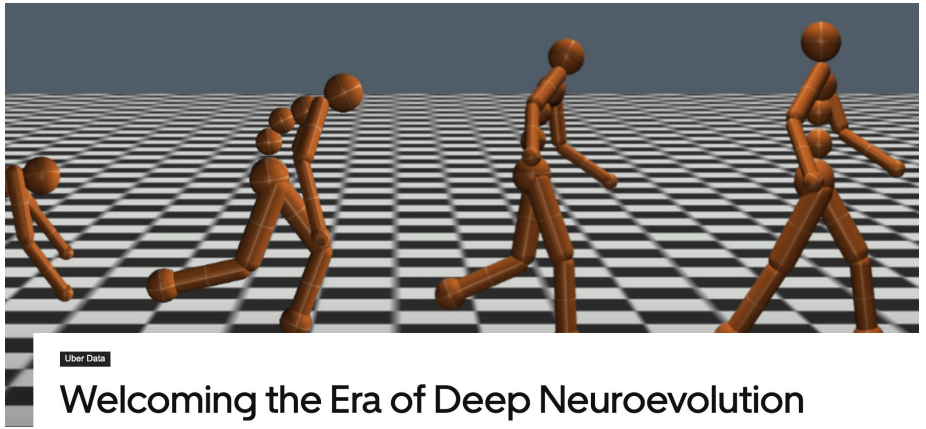
- Pieter Abbeel & Dan Klein’s UC Berkeley CS188 for sharing wonderful resources
- Vladimir Paun
- Photography: Sandy Skoglund

Let’s get started!

https://github.com/NataliaDiaz/IN_104-Projet-Informatique/blob/master/GA_ProjectGuide.pdf

Appendix

References



Intro reads:

- GA: https://en.wikipedia.org/wiki/Genetic_algorithm
- EA: https://en.wikipedia.org/wiki/Evolutionary_algorithm
- Visual Evolution Strategies: http://blog.otoro.net/2017/10/29/visual-evolution-strategies/
- The Era of Deep Neuroevolution (Uber AI Labs): https://eng.uber.com/deep-neuroevolution/
- An intro to genetic algorithms and an example of real industry application on data-driven fashion design
  https://multithreaded.stitchfix.com/blog/2016/07/14/data-driven-fashion-design/

Slides

https://github.com/NataliaDiaz/IN_104-Projet-Informatique

NP-complete problems

- **P** is the set of all problems that can be solved in polynomial time.
- **NP** is the set of all problems whose solutions can be verified in polynomial time
- **NP-complete** is the intersection of **NP-hard** and **NP**
- **NP-hard** is the class of decision problems to which all problems in **NP** can be reduced to in polynomial time by a deterministic Turing machine.