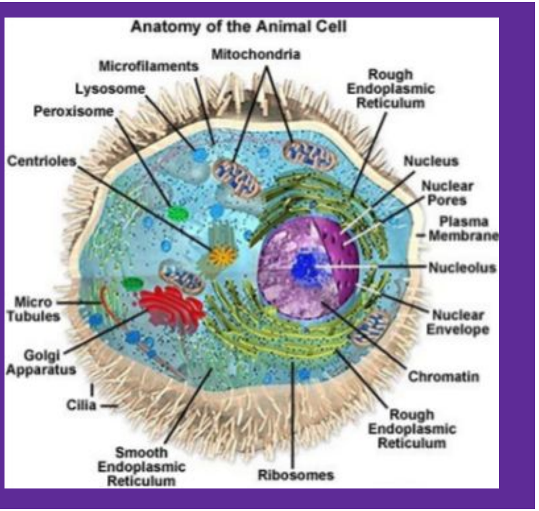
Genetic Algorithms

IN104

Natalia Díaz Rodríguez, ENSTA ParisTech



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

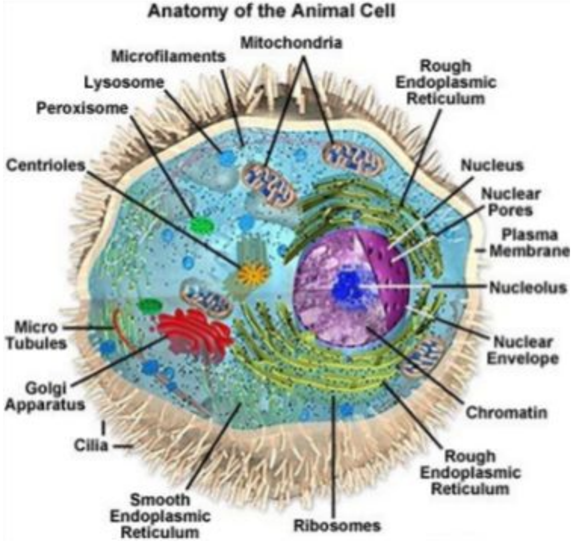
- **EA**: Developed in the 60s
- **GA**: Created by John Holland in the 70s.

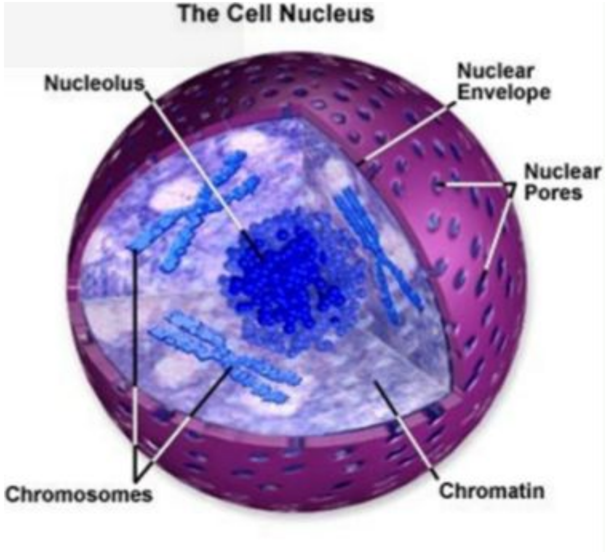
The intuition

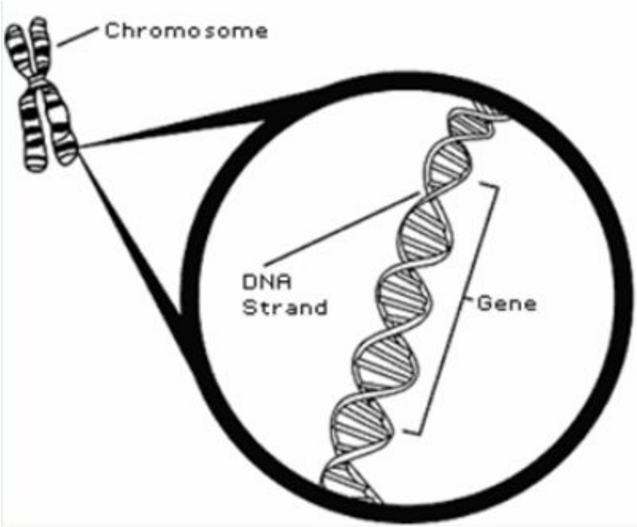
- Based on the model of life beings who are formed by **cells** that contain a **nucleus**
- The genetic information is stored in the **chromosome** that is encoded in the DNA

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**: Recopying existing information
  - **Meiosis** (Reproduction)

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

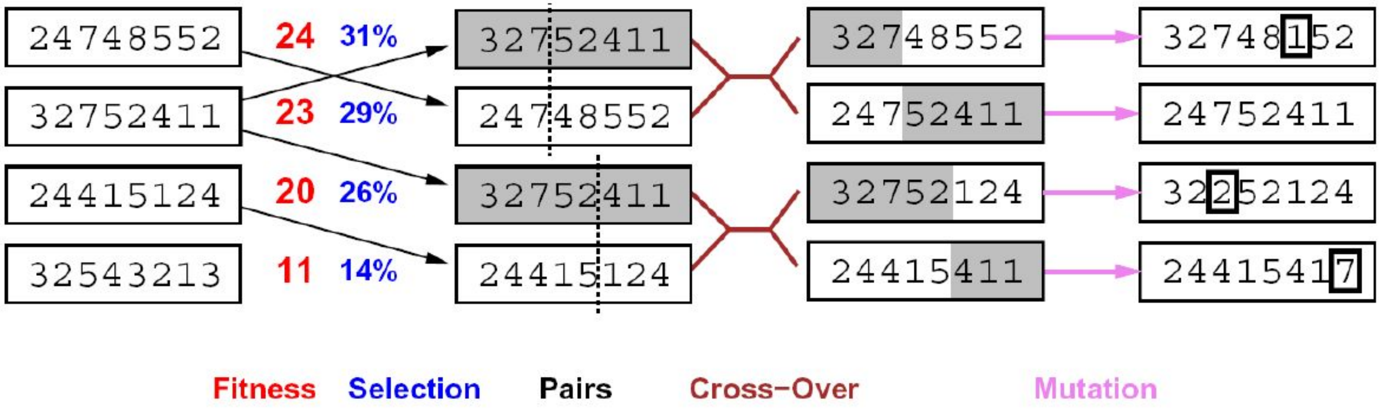
Randomness

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

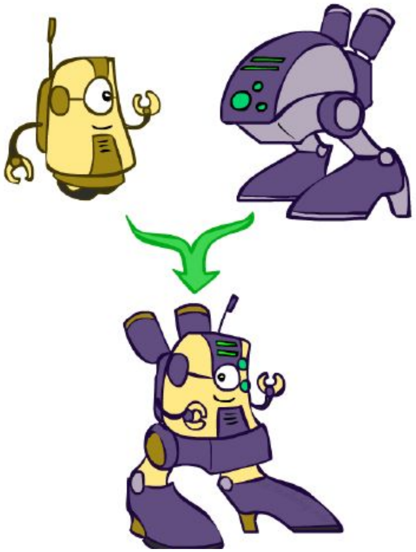
Thanks to these, the population varies to produce a better element, avoiding **local minima**.

=> Mutation

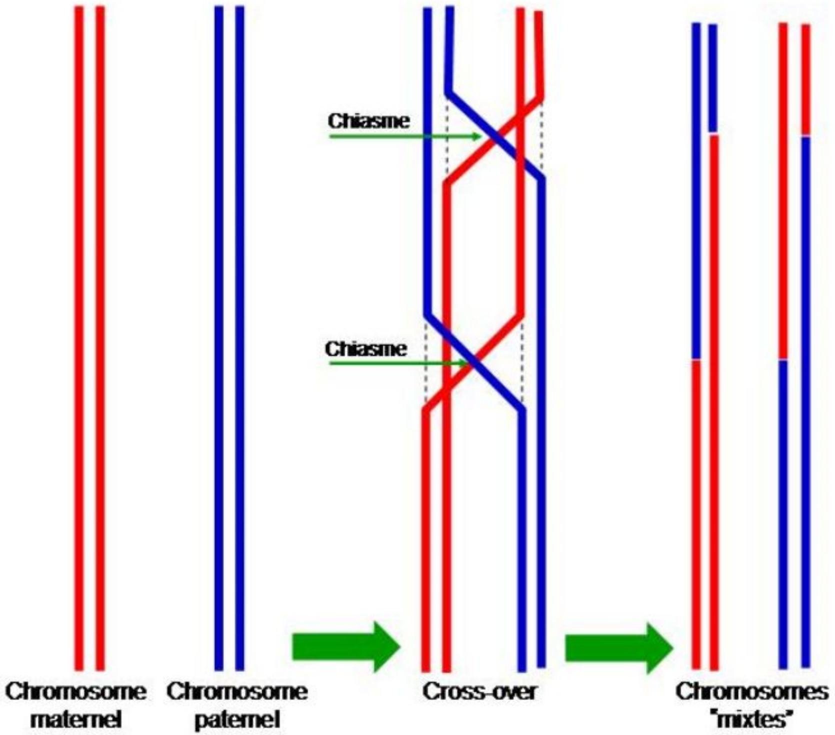
GA Bio-inspired operators:



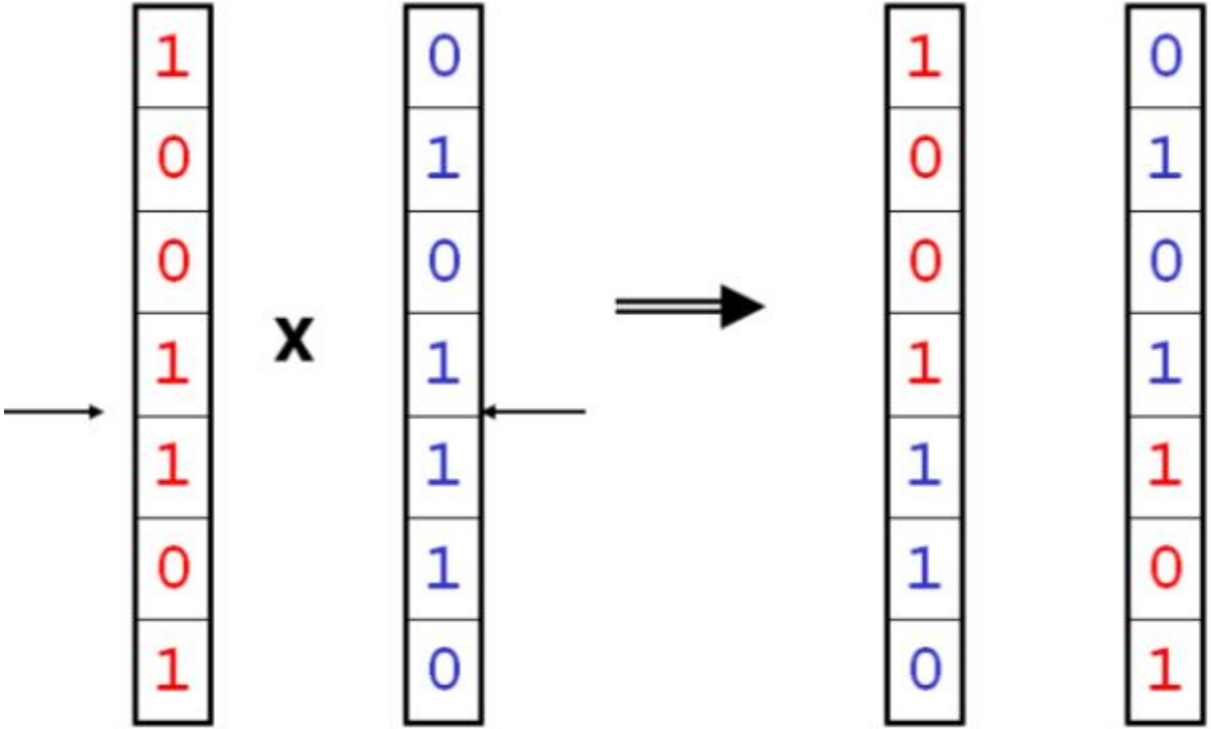
	Fitness	Selection	Pairs	Cross-Over	Mutation
24748552	24	31%	32752411	32748552	32748152
32752411	23	29%	24748552	24752411	24752411
24415124	20	26%	32752411	32752124	32252124
32543213	11	14%	24415124	24415411	24415417



GA Bio-inspired operators: Cross-over

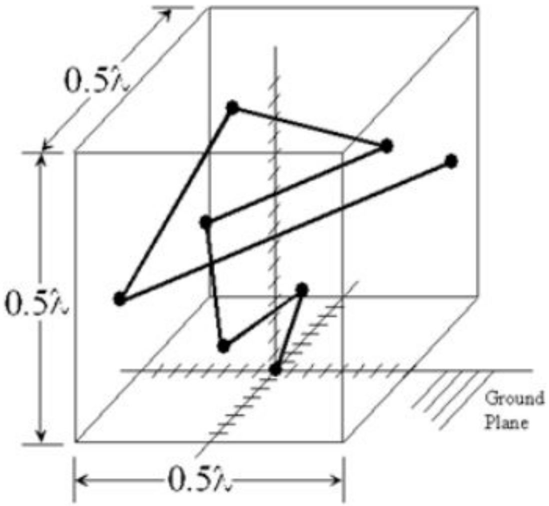


GA Bio-inspired operators: Cross-over



Genetic Algorithms Applications

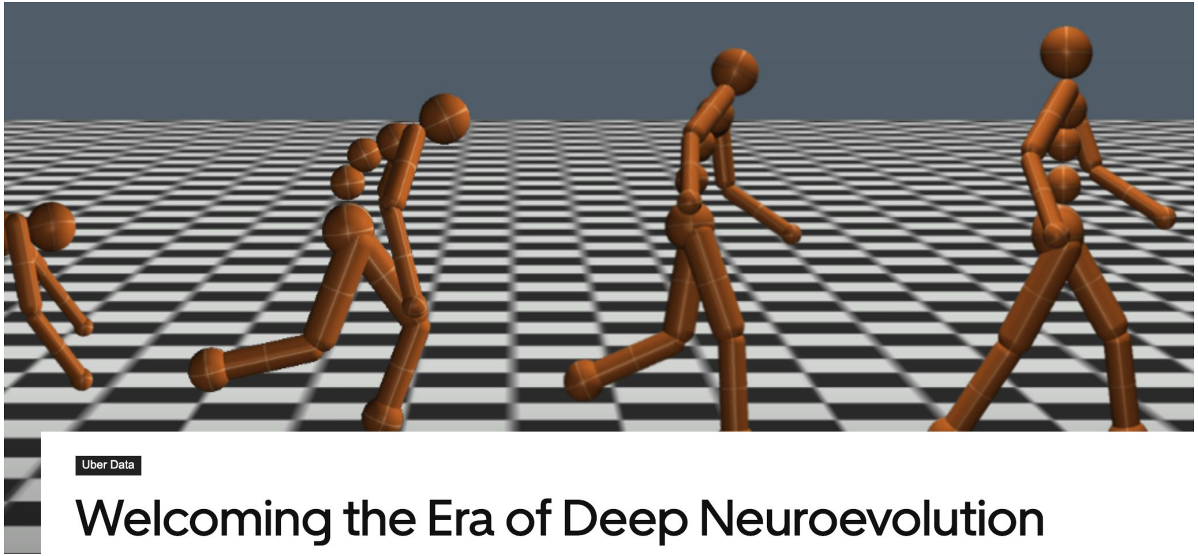
- Finance
  - Stock market data mining optimisation
- Antennas Modelling



- Networks
  - Network intrusion detection

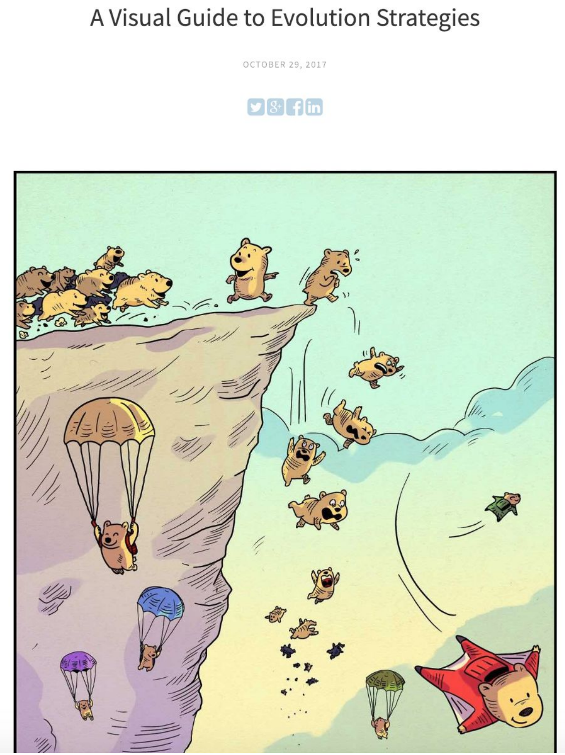
Genetic Algorithms Applications

- Learning locomotion skills, robotics, Atari games…



Uber Data

Welcoming the Era of Deep Neuroevolution

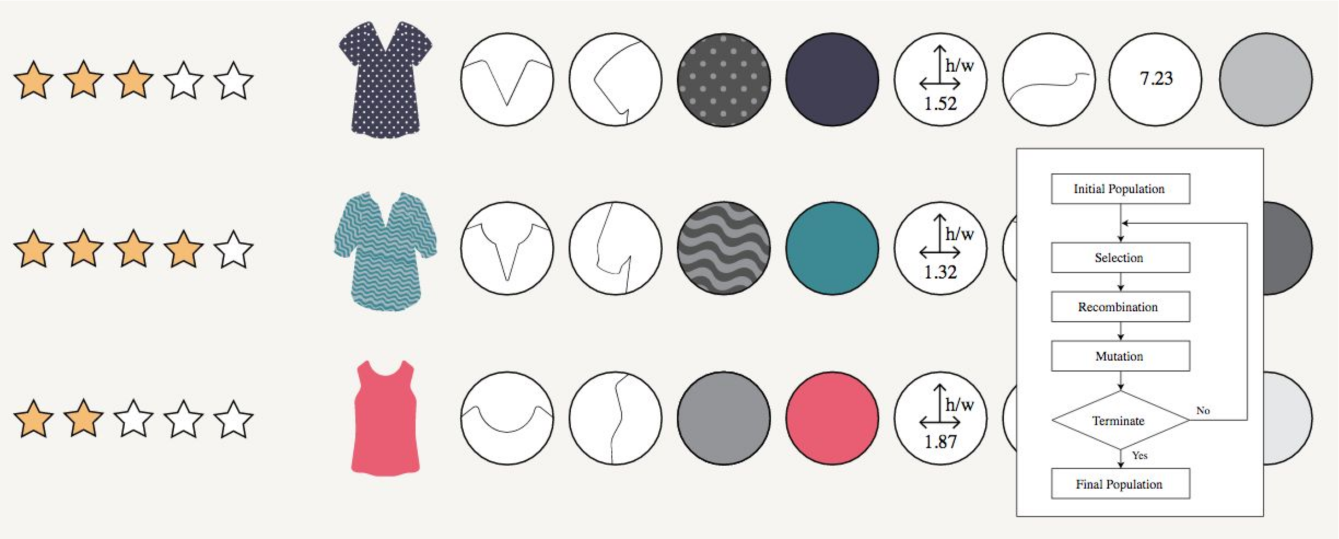


A Visual Guide to Evolution Strategies

OCTOBER 29, 2017

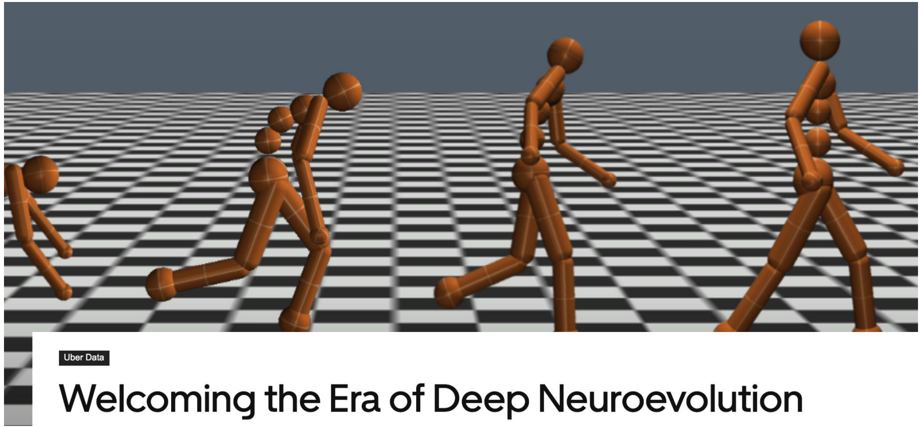
Genetic Algorithms Applications

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



Appendix

References



Welcoming the Era of Deep Neuroevolution

Intro reads:

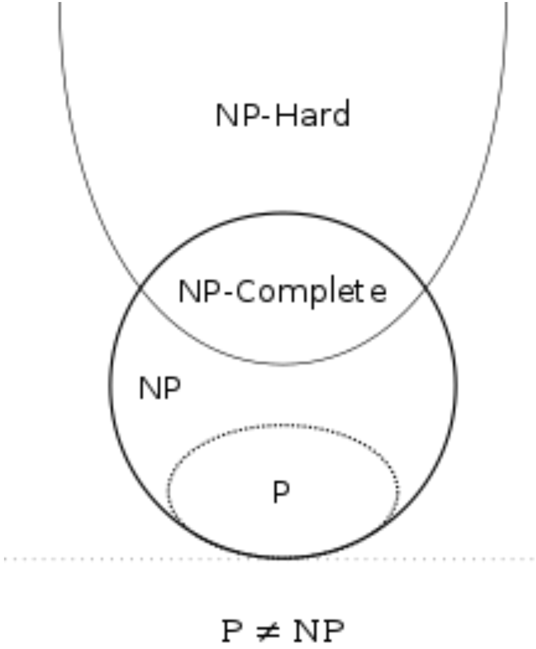
- GA: https://en.wikipedia.org/wiki/Genetic_algorithm
- EA: https://en.wikipedia.org/wiki/Evolutionary_algorithm
- Visual Evolution Strategies: http://blog.otoro.net/2017/10/29/visual-evolution-strategies/
- The Era of Deep Neuroevolution (Uber AI Labs): https://eng.uber.com/deep-neuroevolution/
- An 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.



NP-Hard
NP-Complete
NP
P
$P \neq NP$