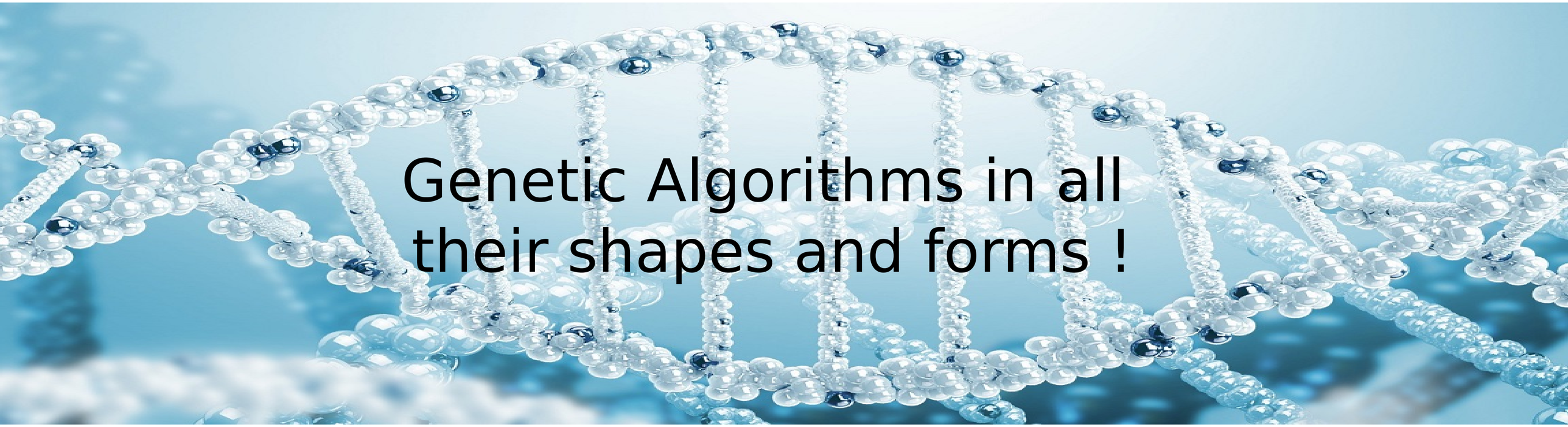
Genetic.io



Genetic Algorithms in all their shapes and forms !

APACHE: BIG_DATA NORTH_AMERICA

Genetic.io
Make something of your big data

Julien Sebrien

- Self-taught, passion for development.
- Java, Cassandra, Spark, JPPF.
- @jsebrien, julien.sebrien@genetic.io

Genetic.io

- Distribution of IT solutions (SaaS, On Premise) allowing the implementation of evolutionary algorithms (genetics, ant colonies, etc.) to optimize business processes.
- Natively distributed architecture.
- Multi-platform (Windows, Unix, Mac), polyglot (Java, Scala, Python, Javascript, R).

Evolutionary computation

Ant colony

Simulated annealing

Differential evolution

Particle swarm optimization

Genetic Algorithms

Memetic Algorithms

And all their variants...

Evolutionary Algorithms

- Inspired by natural physical mechanisms.
- Generate high-quality solutions for optimization and search problems by relying on bio-inspired operators such as mutation, crossover and selection.

Applications

Marketing

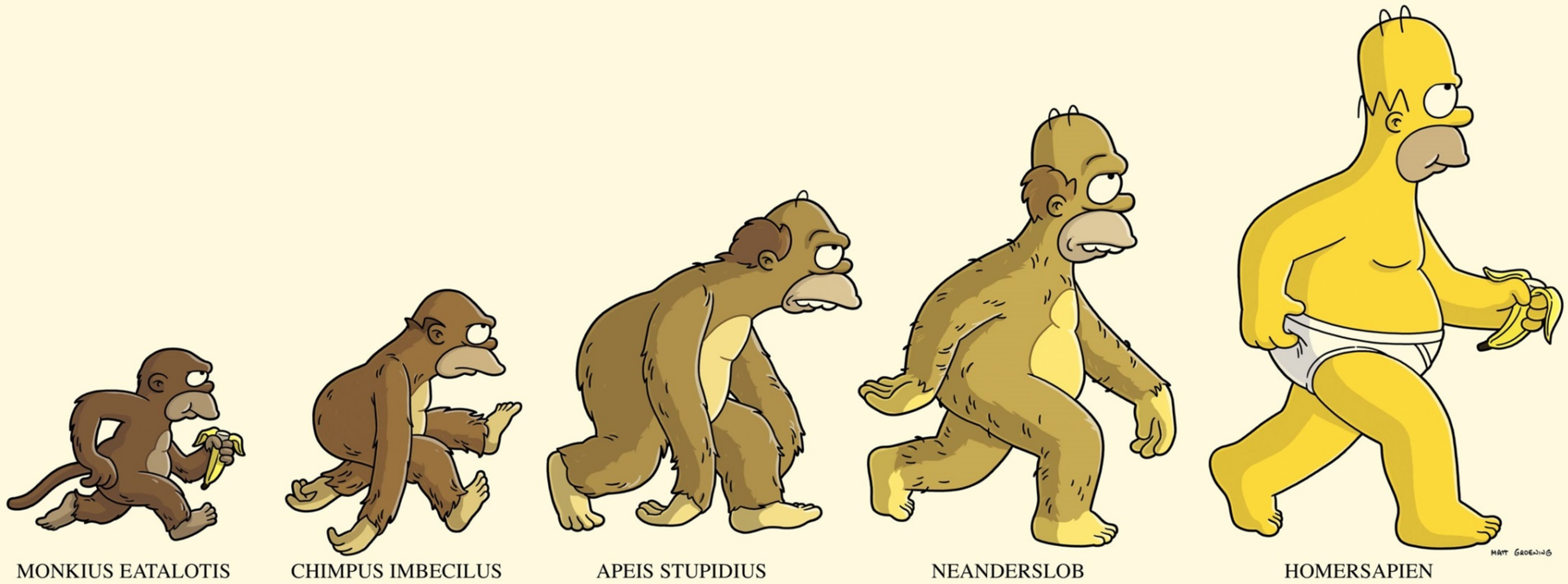
Tourism marketing :
https://goo.gl/aCc9SJ

Astronautics

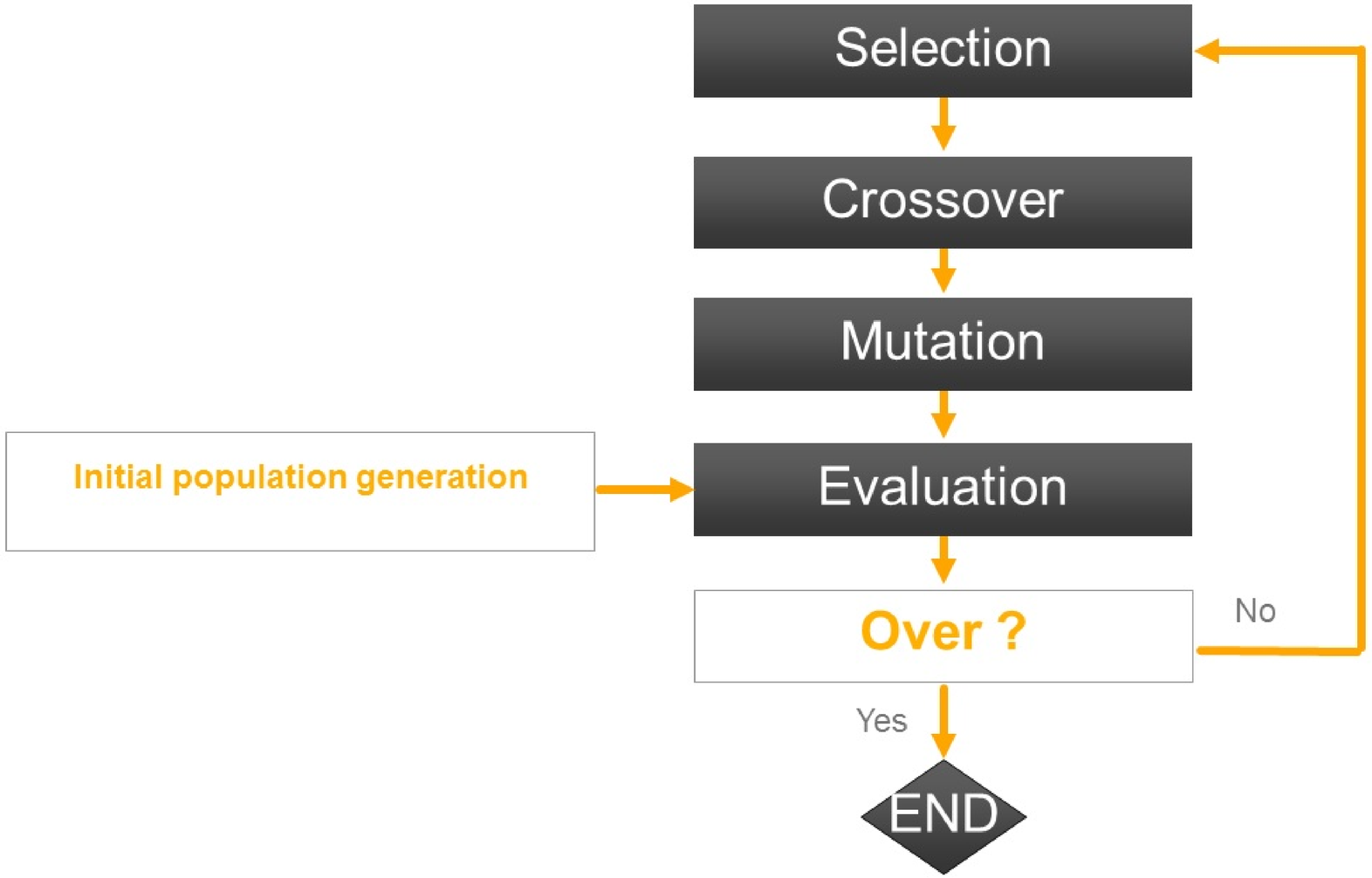
Genetic algorithms 'naturally select' better satellite orbits :
https://goo.gl/eauC32

And others : Imagery, Linguistics: https://en.wikipedia.org/wiki/List_of_genetic_algorithm_applications

Genetic Algorithms



Workflow


Selection
Crossover
Mutation
Initial population generation
Evaluation
Over ?
No
Yes
END

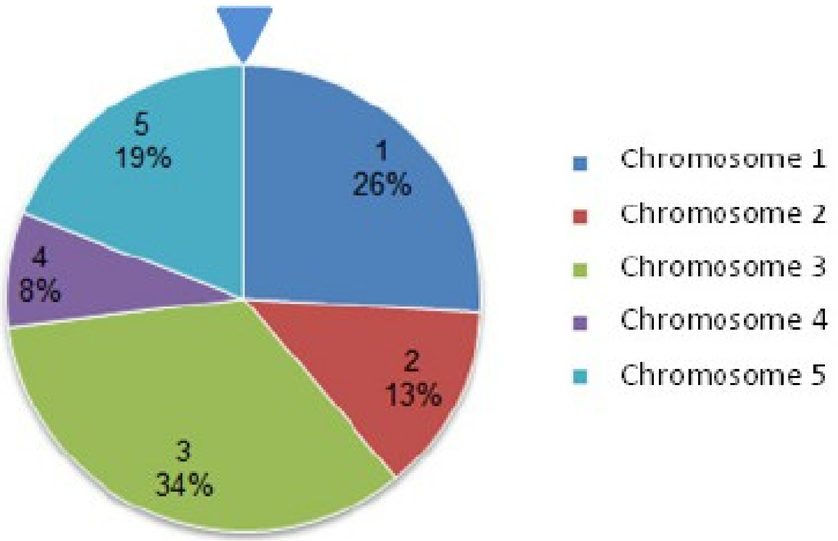
Selection

Several selection method exist: Roulette Wheel, Tournament, Ranking, etc.

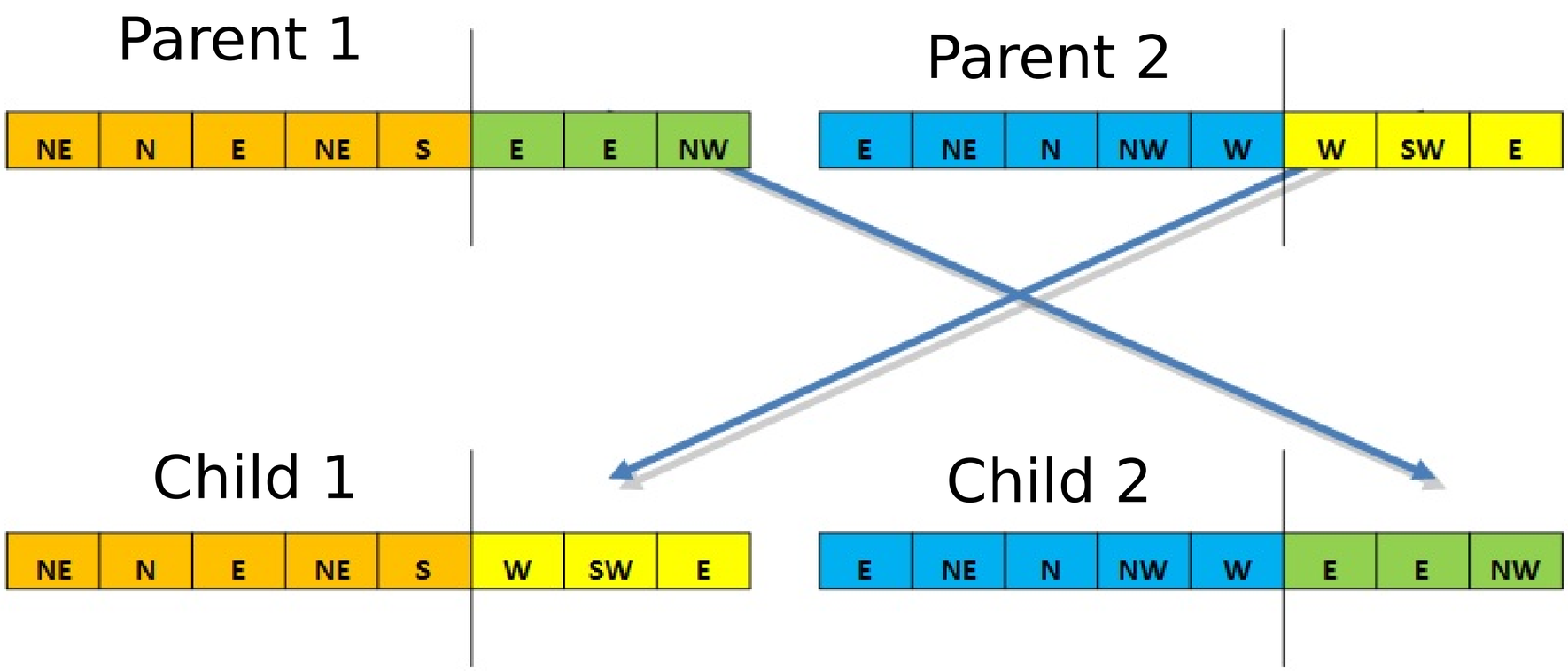
Roulette Wheel example :

- Parents are selected according to their fitness.
- The fittest individuals have a greater chance of survival than weaker ones.
- Selection probability:

$$p_i = \frac{f_i}{\Sigma_{j=1}^{N} f_j}$$

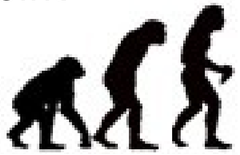


Crossover



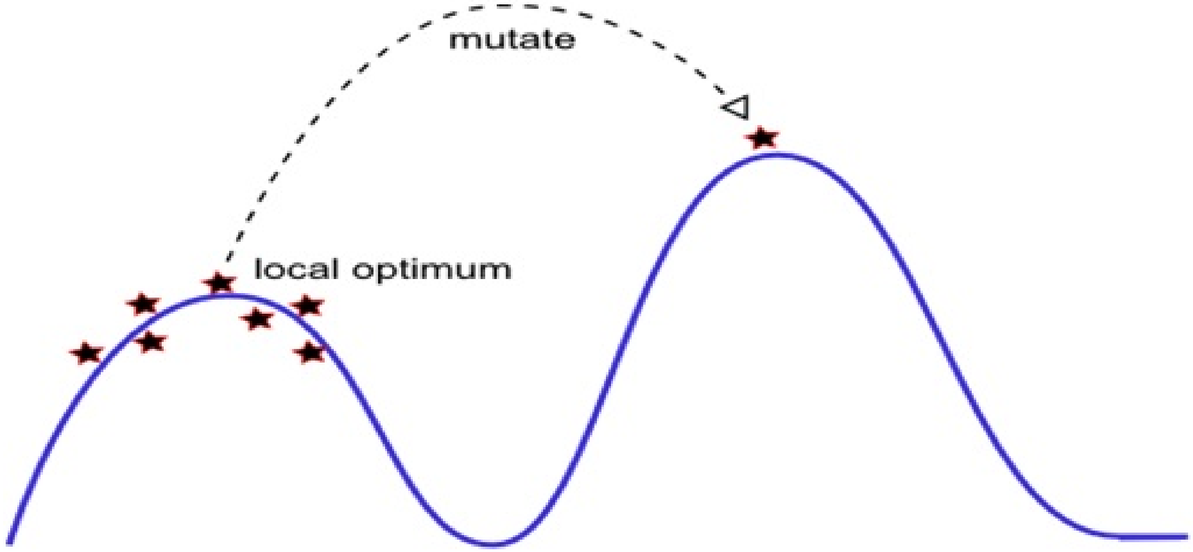
1 crossover point

1 crossover point

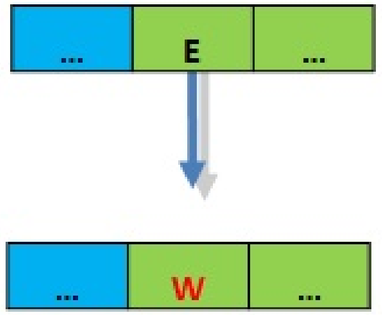


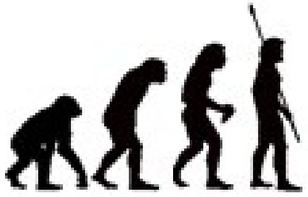
Mutation

- Injects diversity into the population, reducing the risk of stagnation within a local optimum.
- Mutation rate generally between 1 and 5%.



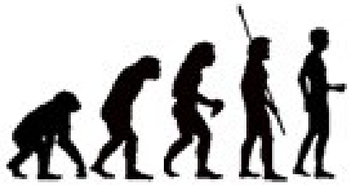
child





Evaluation

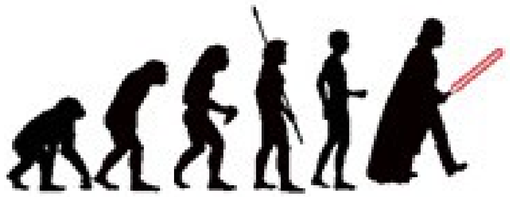
- Takes a candidate solution to the problem as input and produces as output how “fit” or how “good” the solution is with respect to the problem in consideration.
- The assigned score is ideally independent of other individuals in the population.
- This function should be carefully implemented in order to increase the probability of convergence of the algorithm.



Termination

The algorithm ends if one of the following termination conditions is satisfied :

- A maximum number of generations is reached.
- A candidate has a fitness score greater than or equal to a previously defined threshold.
- The algorithm has been running for too long.
- Etc.




#ToBeOrNotToBe

« TOBEORNOTTOBE » use case

Modelization:

- Initial genome consisting of a sequence of 13 characters, generated randomly.

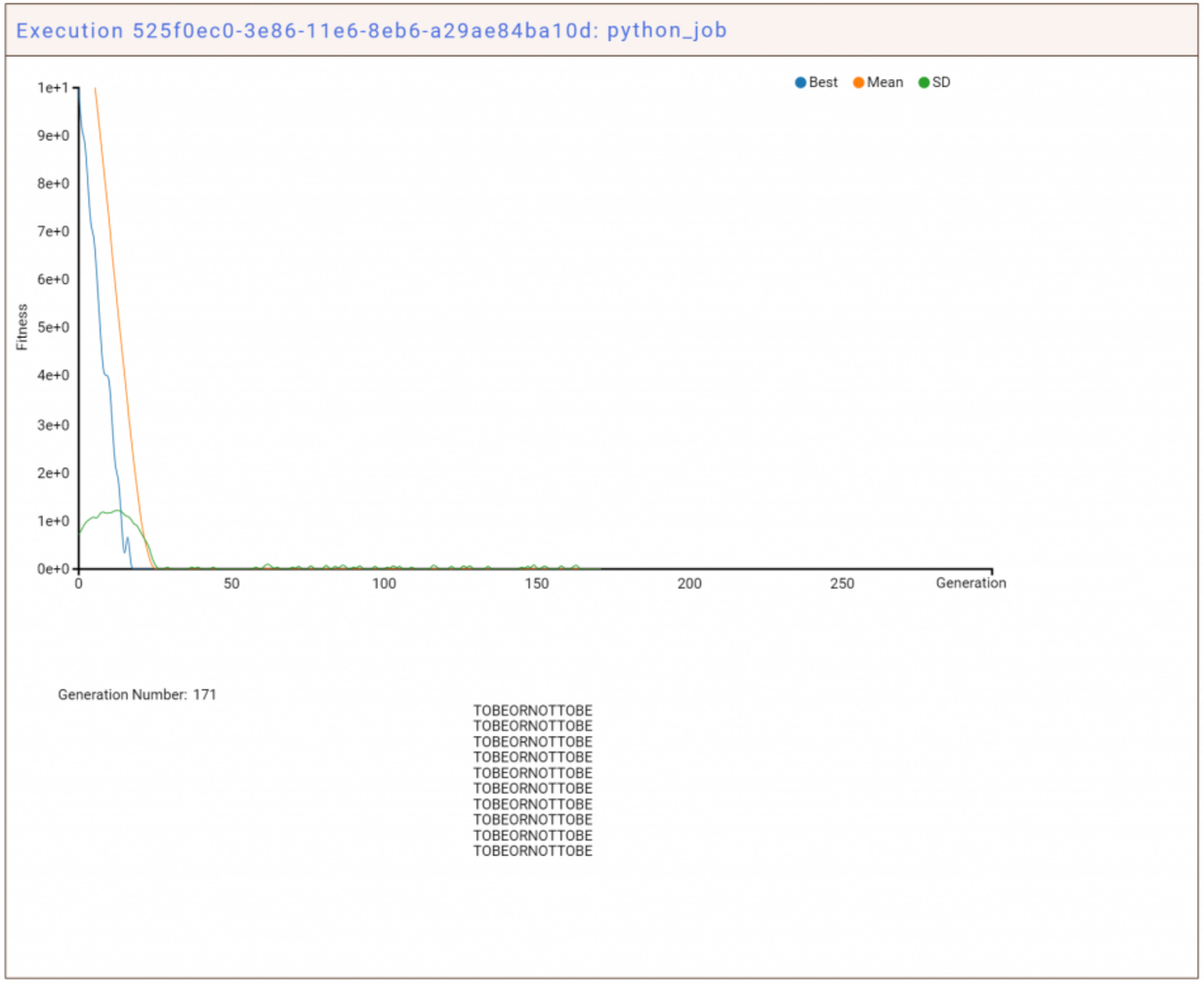
Fitness function:

- Sum of the differences between the letter of the genome and the target letter, at each position:

Genome	C	Q	Y	T	C	Z	K	I	H	U	E	I	T
Target	T	O	B	E	O	R	N	O	T	T	O	B	E
Gap (absolute value)	17	2	23	15	12	8	3	6	12	1	10	7	15

Score = 131

Execution


Execution 525f0ec0-3e86-11e6-8eb6-a29ae84ba10d: python_job
Best
Mean
SD
Fitness
1e+1
9e+0
8e+0
7e+0
6e+0
5e+0
4e+0
3e+0
2e+0
1e+0
0e+0
0
50
100
150
200
250
Generation
Generation Number: 171
TOBEORNOTTOBE
TOBEORNOTTOBE
TOBEORNOTTOBE
TOBEORNOTTOBE
TOBEORNOTTOBE
TOBEORNOTTOBE
TOBEORNOTTOBE
TOBEORNOTTOBE
TOBEORNOTTOBE
TOBEORNOTTOBE

« Smart Rockets » use case

« Smart Rockets » use case

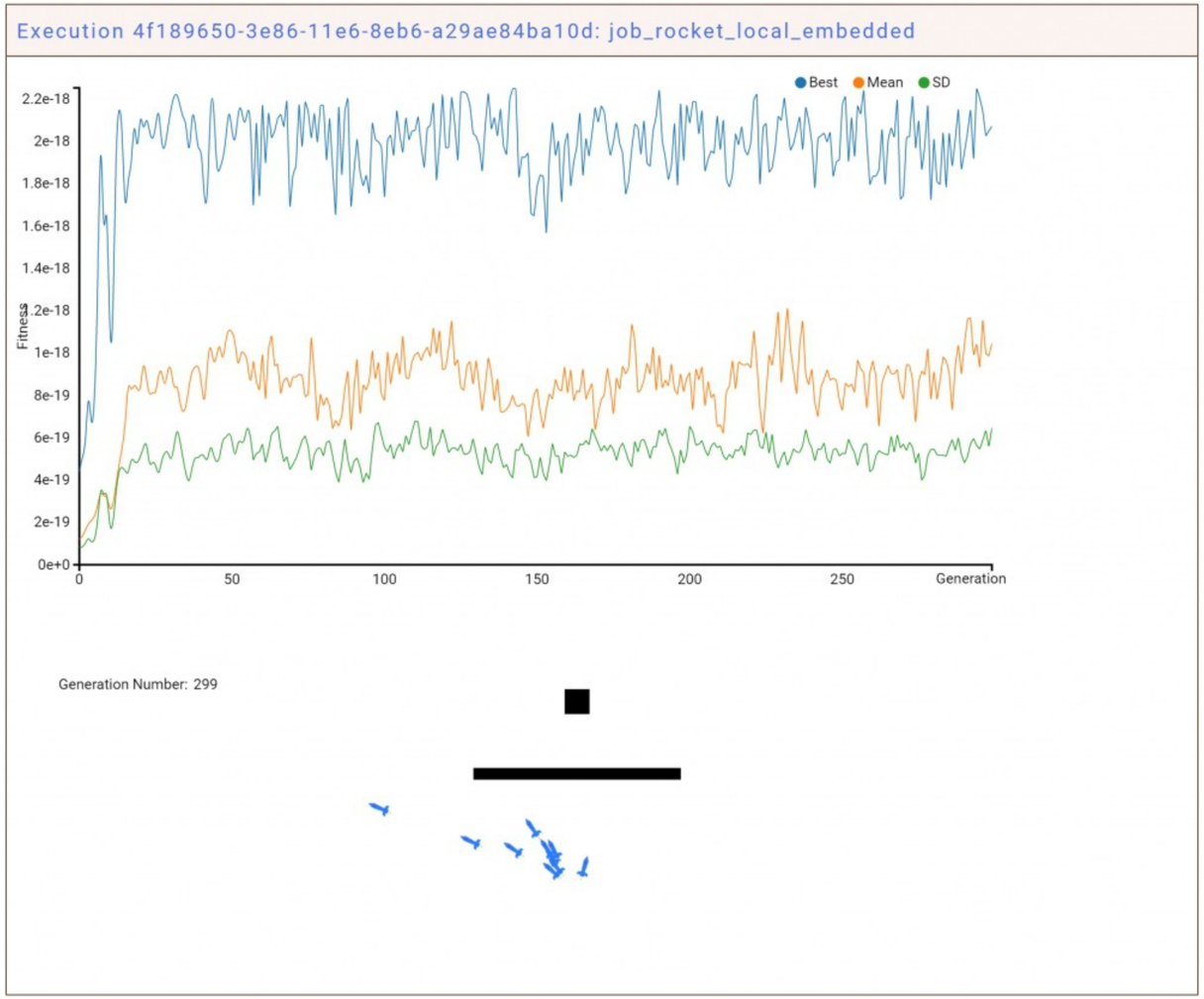
Modelization:

A sequence of 300 acceleration vectors in a 2D plane.

Fitness function:

- The closer an individual is to his target at the end of his movement, the higher will be his score.
- An individual's score will be severely penalized if he touches the obstacle during his movement.
  - Score = 1/ R (with R = remaining distance from target) ; If Obstacle hit, Score = Score / 10 !

Execution


Execution 4f189650-3e86-11e6-8eb6-a29ae84ba10d: job_rocket_local_embedded
Best
Mean
SD
Fitness
2.2e-18
2e-18
1.8e-18
1.6e-18
1.4e-18
1.2e-18
1e-18
8e-19
6e-19
4e-19
2e-19
0e+0
0
50
100
150
200
250
Generation
Generation Number: 299

Tourist sites location optimization

Selection of tourist sites

Modelization:

- A sequence of bits, representing the presence of an agency on a site: 010100001

Selection of tourist sites

Fitness function:

- Depends on the following variables:

  - The number of tourist sites.
  - The number of competitors in the territory in question.
  - The number of geographical areas within the territory concerned.
  - The number of categories of households.
  - The average expenditure of an individual in category k, for a tourist product or service.
  - The number of k categories of households located in an area i.
  - Subjective measures of attraction of a site j.
  - The subjective measures of attraction of the services offered on each site.
  - The transport time from an area i to a site j.
  - Transport time from an area i to an existing tourist location.

Selection of tourist sites

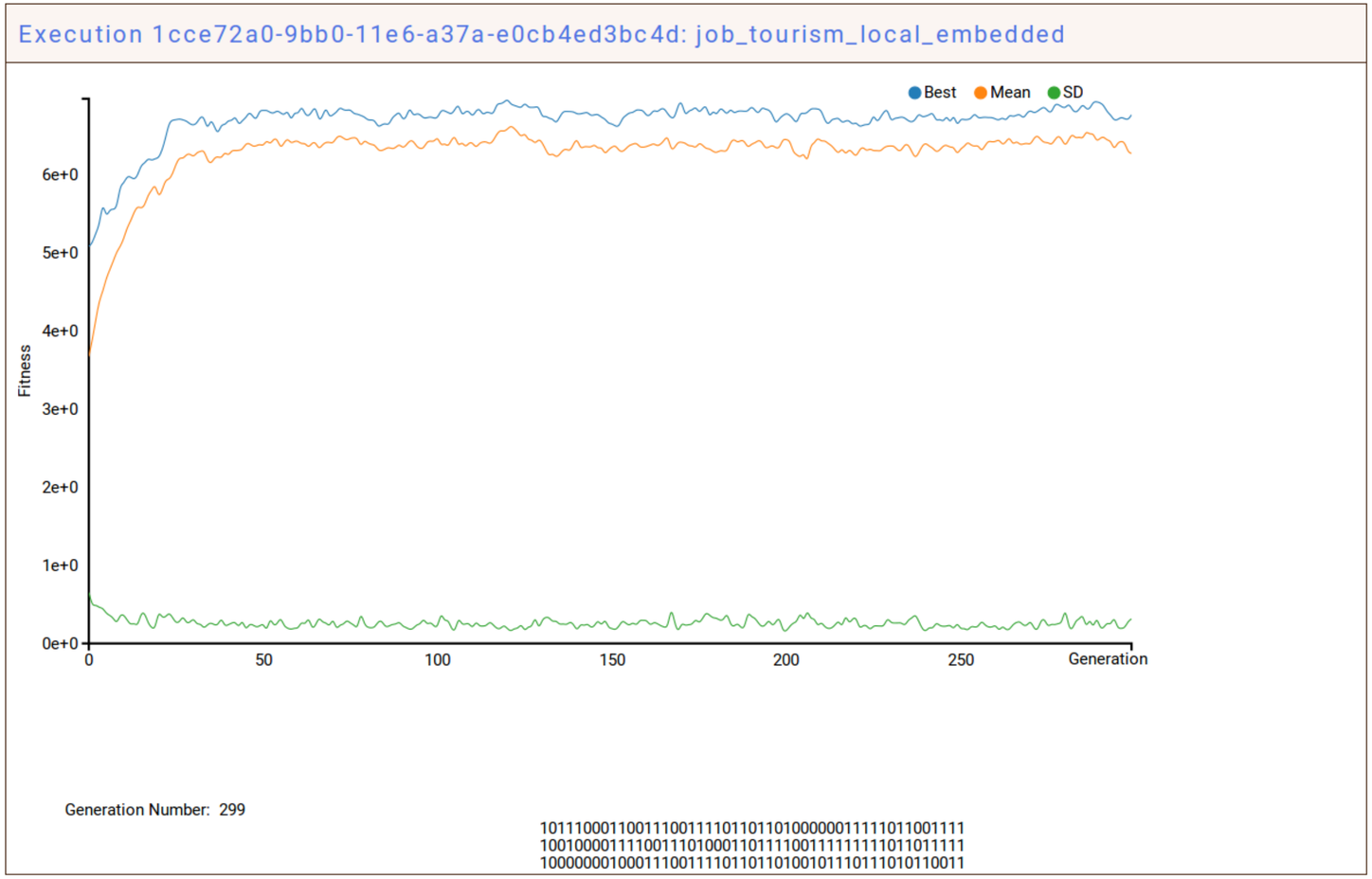
$$EXP_{ij} = \beta_0 \left(\sum_{k=1}^{N_k} E_k H_{ik} \right) \cdot \frac{W_j e^{-\beta_1 T_{ij}}}{\sum_{m=1;m\neq j}^{N_s+N_c} W_m e^{-\beta_1 T_{im}}}$$

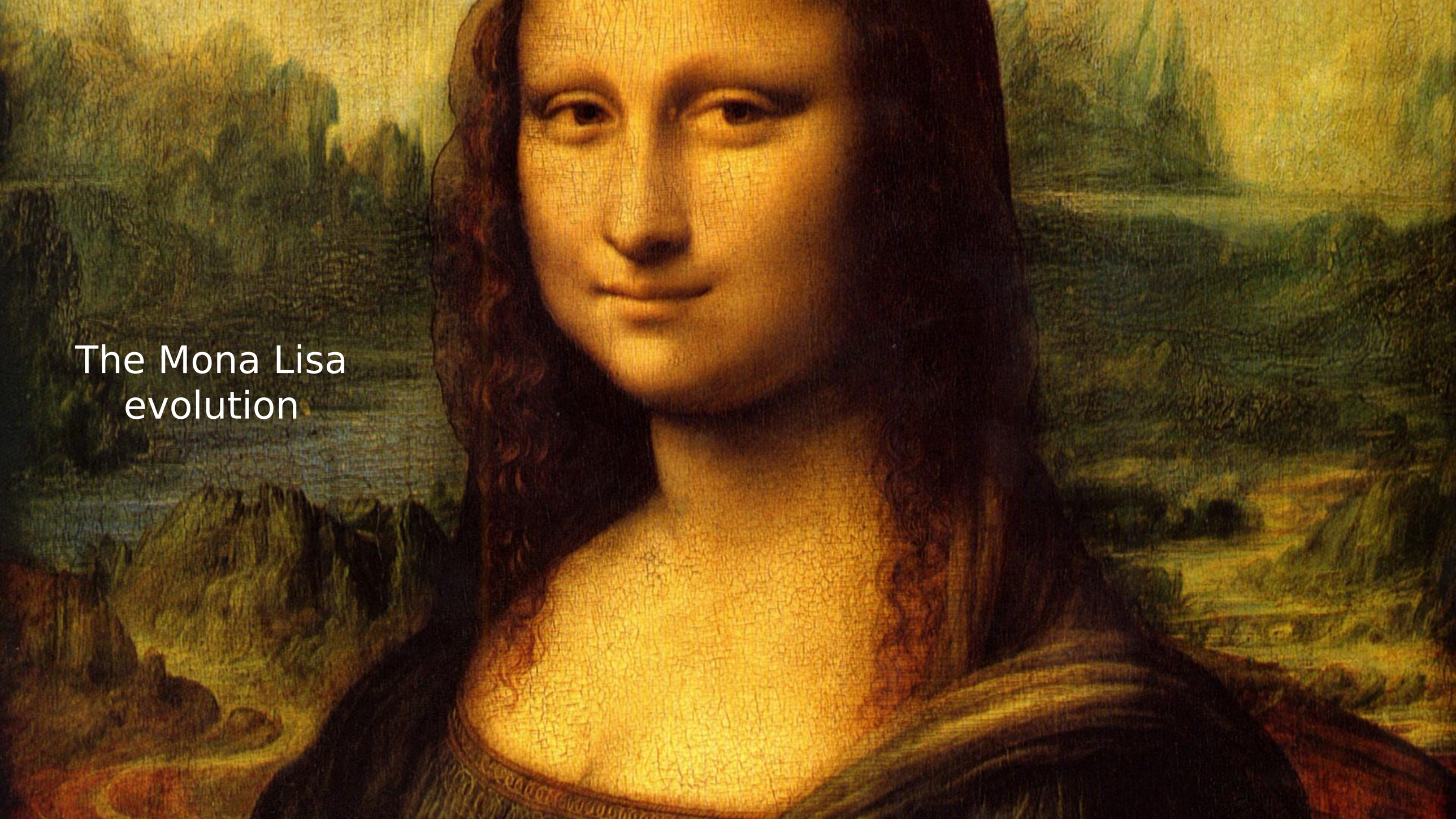
$i = 1, 2, ..., N_z$
$j = 1, 2, ..., N_s$

EXP_{ij} is tourism expenditure from zone i at site j
β_0 and β_1 are parameters
N_k is the number of household categories
$N_s = S_e$ (number of existing sites) $+ S_p$ (number of potential new sites)
N_c is the number of competitors
N_z is the number of zones
E_k is mean expenditure on the tourism product/service category by house hold category k
H_{ik} is the number of households in category k located in zone i
W_j is a subjective measure of attraction of site j
W_m is a subjective measure of attraction of existing/proposed tourism outlets
$(m = 1, 2, \ldots, N_s)$ and those of competitors $(m = N_s + 1, ..., N_s + N_c)$
T_{ij} is the travel time from zone i at site j
T_{im} is the travel time from zone i to existing/proposed tourism sites $(m = 1, 2, ..., N_s)$
and those of competitors $(m = N_s + 1, ..., N_s + N_c)$

Objective: maximize $\sum \sum EXP_{ij}$

Execution


Execution 1cce72a0-9bb0-11e6-a37a-e0cb4ed3bc4d: job_tourism_local_embedded
Best
Mean
SD
Fitness
6e+0
5e+0
4e+0
3e+0
2e+0
1e+0
0e+0
0
50
100
150
200
250
Generation
Generation Number: 299
1011100011001110011110110110110100000011111011001111
1001000011110011101000110111100111111111011011111
1000000010001110011110110110100101110111010110011


The Mona Lisa
evolution

The Mona Lisa evolution

Method:

- Random generation of polygons

DNA:

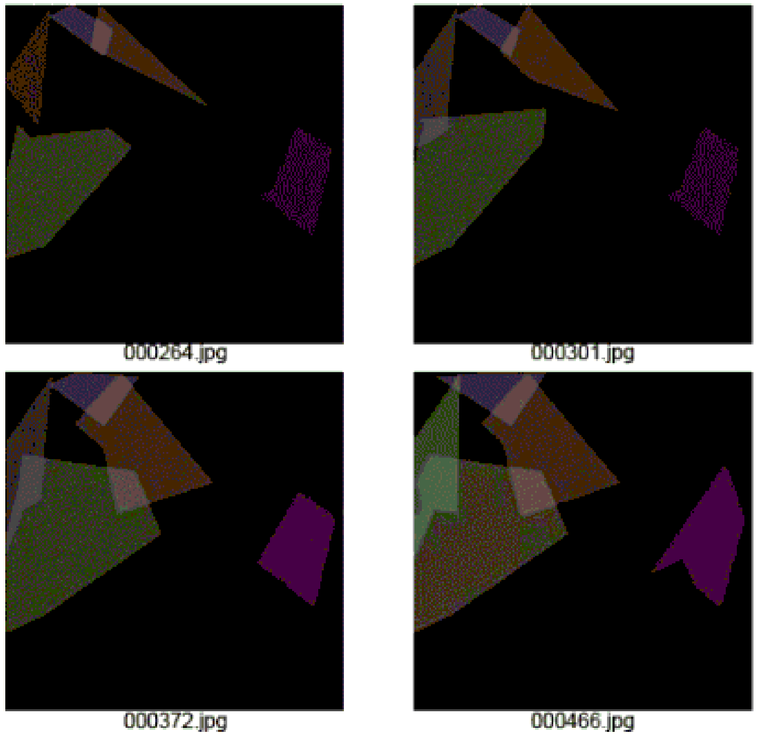
- Image pixels (int array)

Fitness:

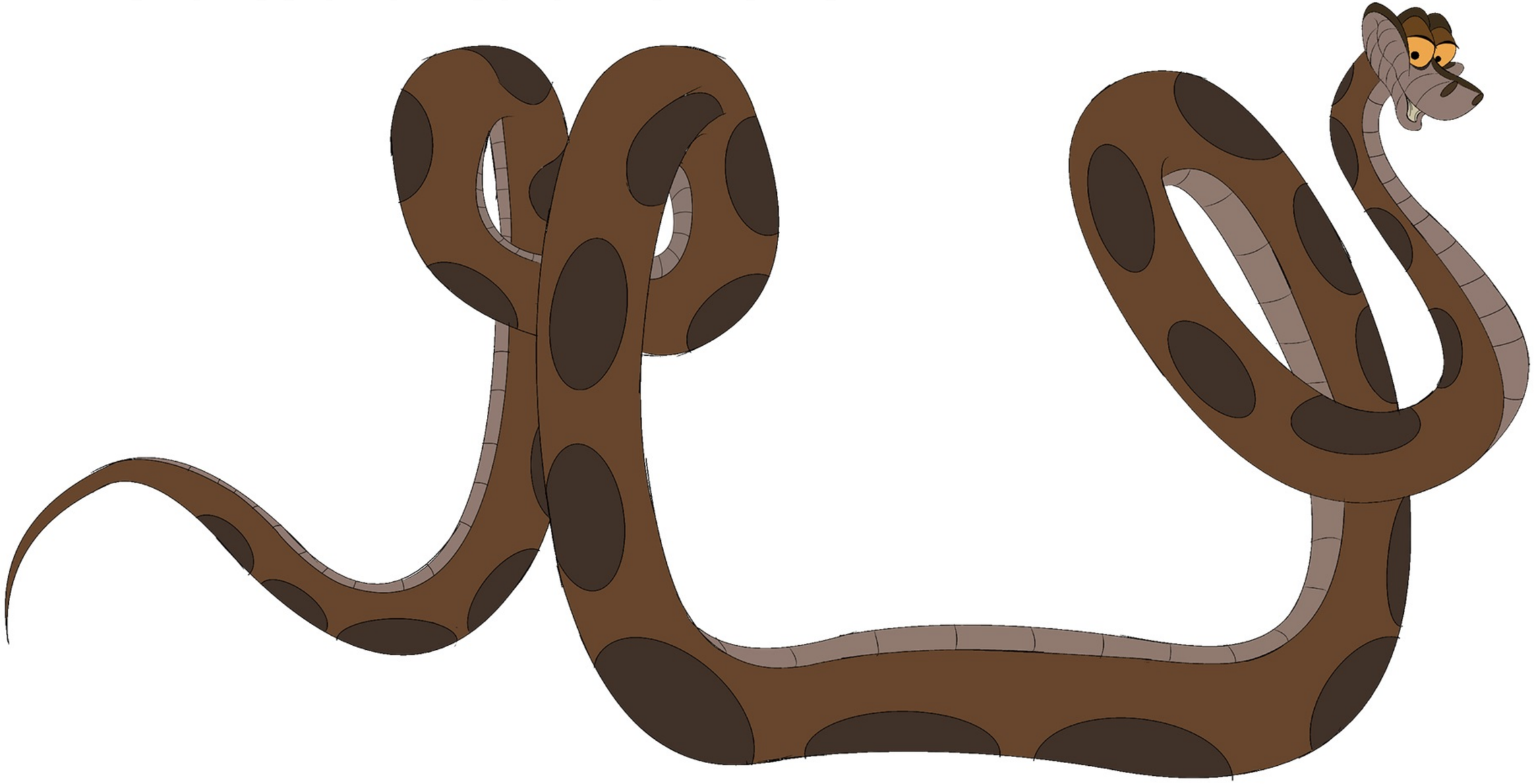
- Calculation of color differences

for each component (R,G,B)

Score = $\sum_{i=0}^{nbPixels} \left(\sqrt{\triangle R_i^2 + \triangle G_i^2 + \triangle B_i^2} \right)$



Snakes and neural networks



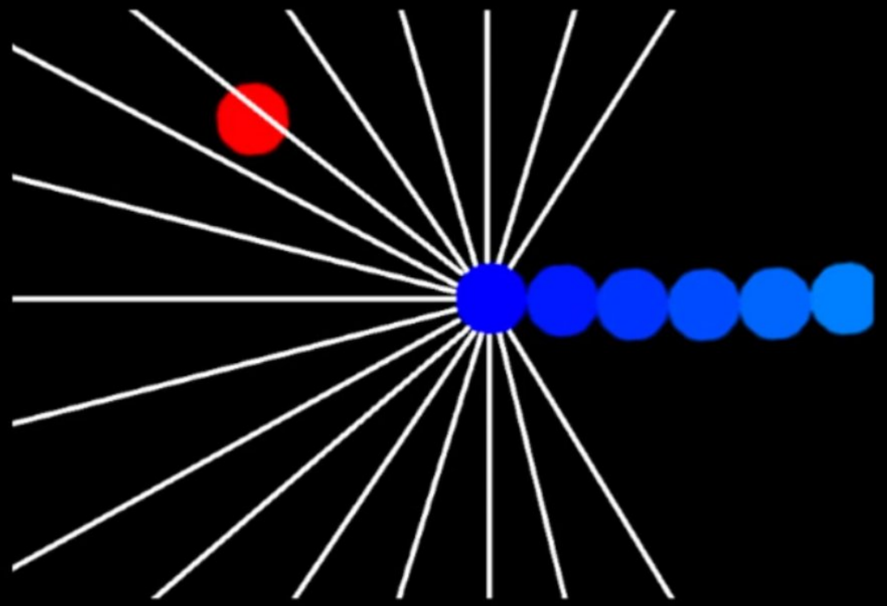
Snakes and neural networks

Vision 240 degrees, divided into 16 parts

Three types of objects encountered:

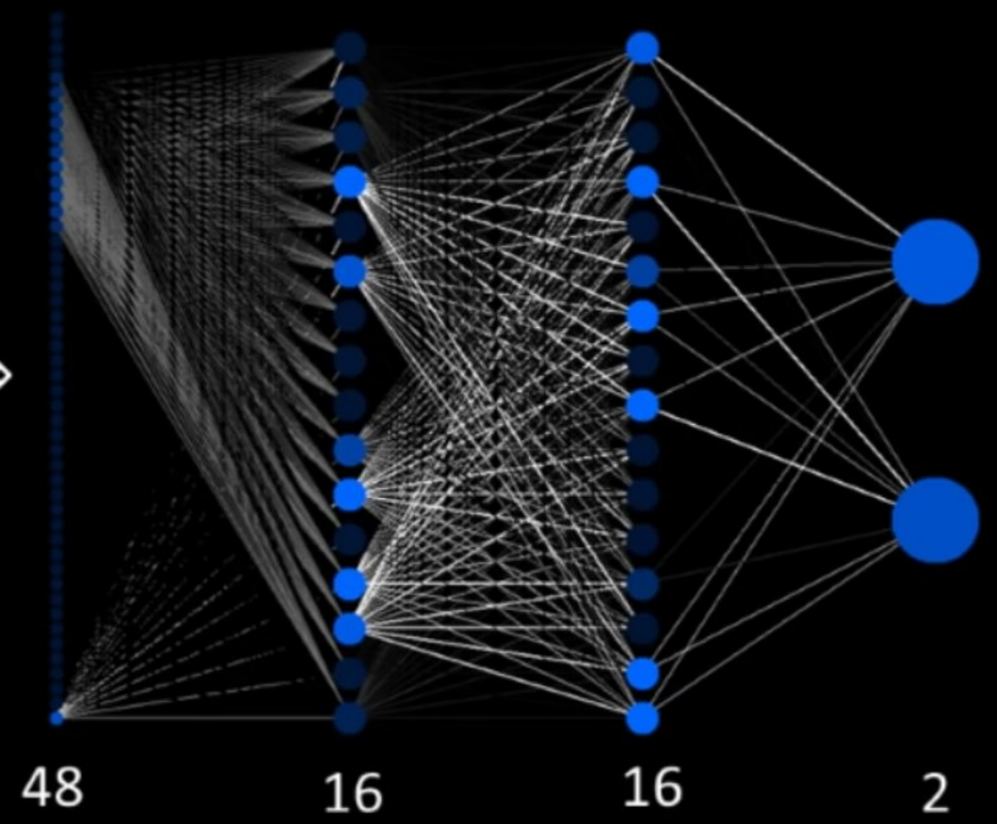
- Wall
- Food
- Himself!

=> 48 input neurons



Snakes and neural networks

Inputs ⇒



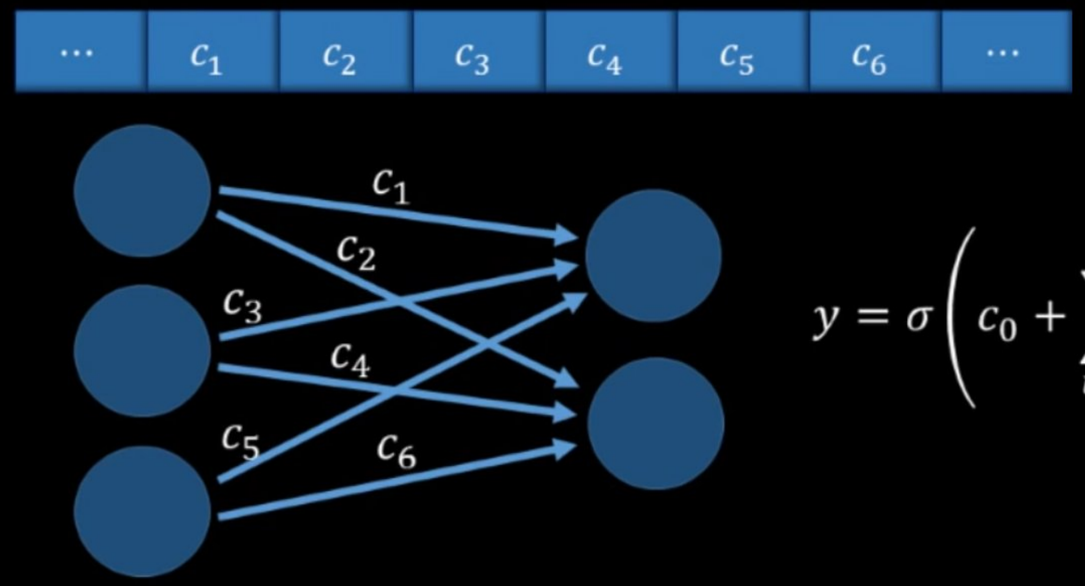
⇒Outputs (angle = output 1 – output 2)

$$s = c_0 + \sum_{i=1}^{N} c_i x_i$$

$$y = \frac{1}{1 + e^{-x}}$$

Snakes and neural networks

DNA :



$$y = \sigma\left(c_0 + \sum_{i=1}^{N} c_i x_i\right)$$

Fitness : 20 . Size + 5 . Health (ie. Dead or Alive!)

Antenna of NASA's ST5 mission

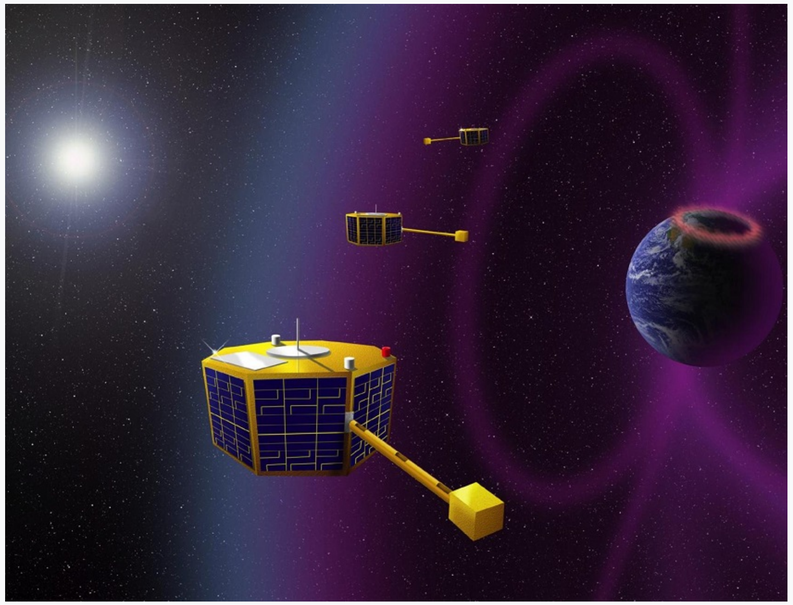
Antenna of NASA's ST5 mission

Objective:

- Optimize the design and efficiency of antennas placed on orbiting satellites

Constraints:

- VSWR < 1,2 (Voltage Standing Wave Ratio) at transmission frequency (8470 Mhz), VSWR < 1,5 at reception fréquency (7209,125 Mhz)
- Input Impedance at 50 Ohms
- Mass < 165g, contained in cylinder of diameter / height < 15,24 cm



Antenna of NASA's ST5 mission

Fitness function:

Minimize: $$F = VSWR \cdot gain_{error} \cdot gain_{smoothness}$$

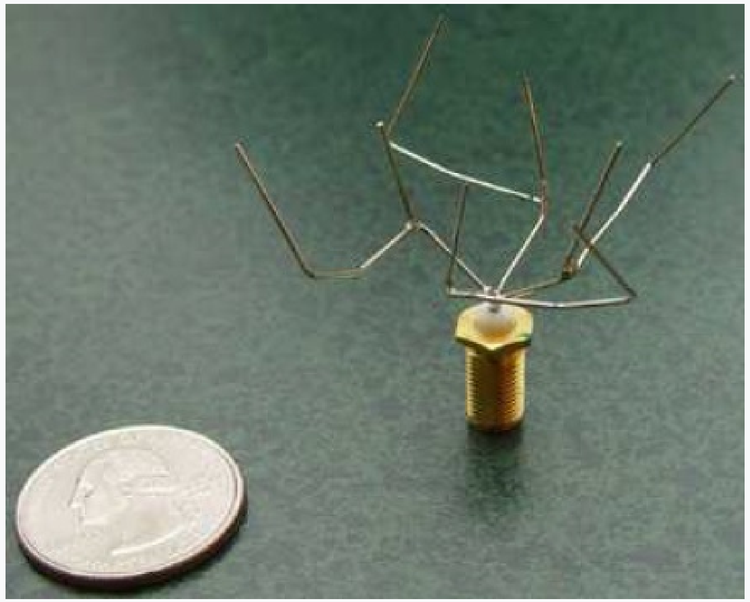
With:

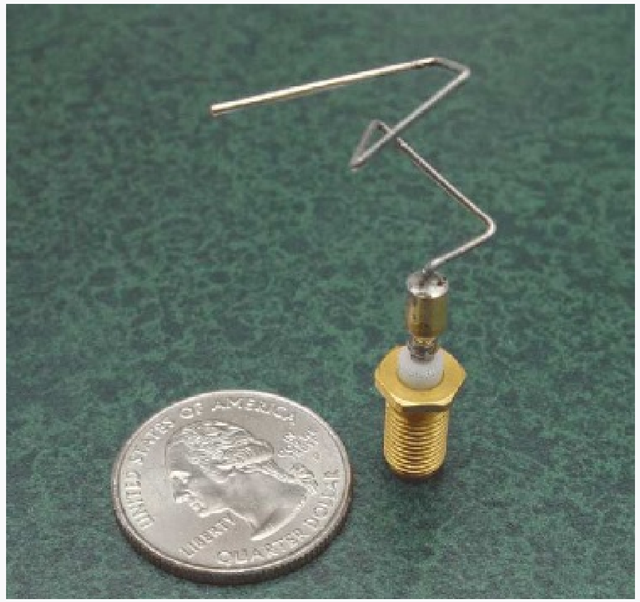
$VSWR$ (Voltage Standing Wave Ratio) = $\frac{1+|\Gamma|}{1-|\Gamma|}$, $\Gamma = \frac{V_r}{V_f}$ (« reflected wave », « forward wave »)

$gain_{error}$: the sum of the penalties calculated for each angle

$gain_{smoothness}$: take into account the smoothing of gains

Antenna of NASA's ST5 mission

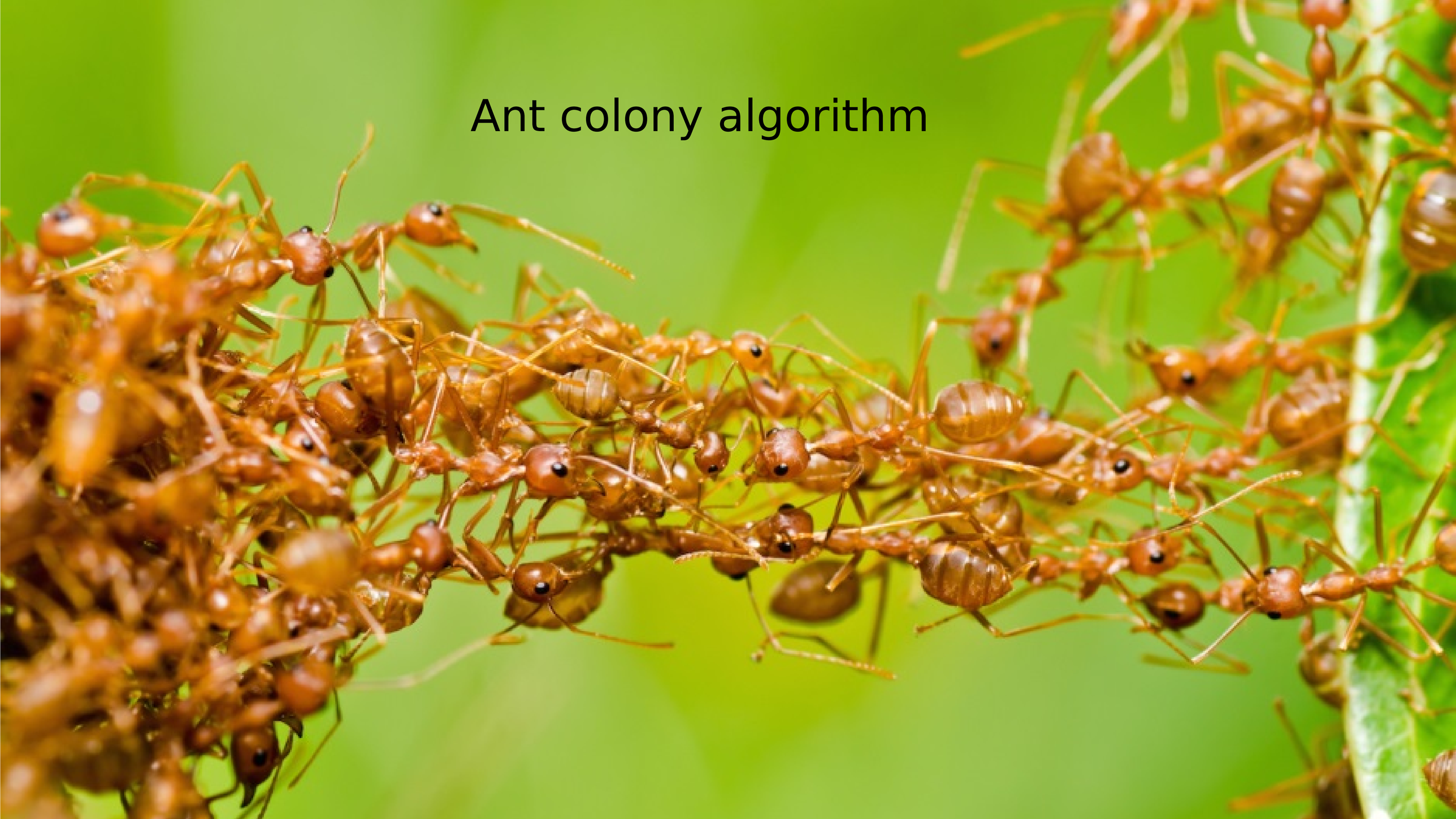




Advantages :

- Better consumption, manufacturing time, performance, complexity, reliability
- Improved efficiency by 93%, for 2 QHA (quadrifilar helix antenna)

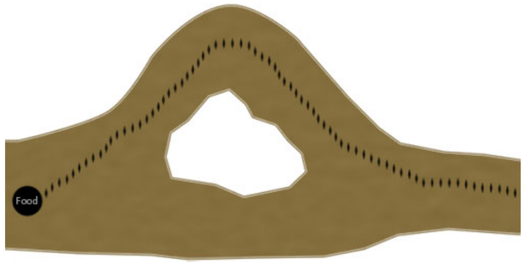
Ant colony algorithm



Ant colony algorithm

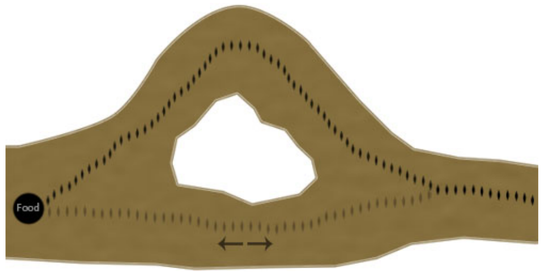
- the ant can only visit each city once
- the more a city is far away, the less likely it is to be chosen ("visibility")
- the greater the intensity of the pheromone trail on the ridge between two cities, the more likely the path will be chosen
- once the path has been completed, the ant deposits more pheromones on all the edges traveled if the path is short
- the pheromone tracks evaporate at each iteration

Ant colony algorithm

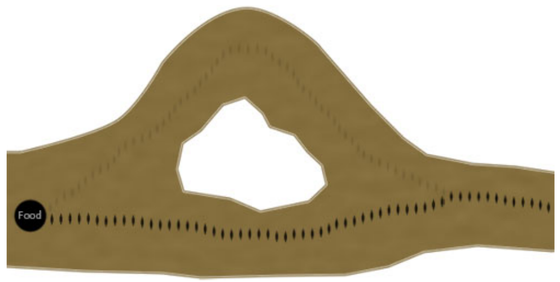


Initial Path

New Pheromone Reinforced Path



Path finally adopted by the colony



Application to the travelers salesman problem!

Simulated annealing

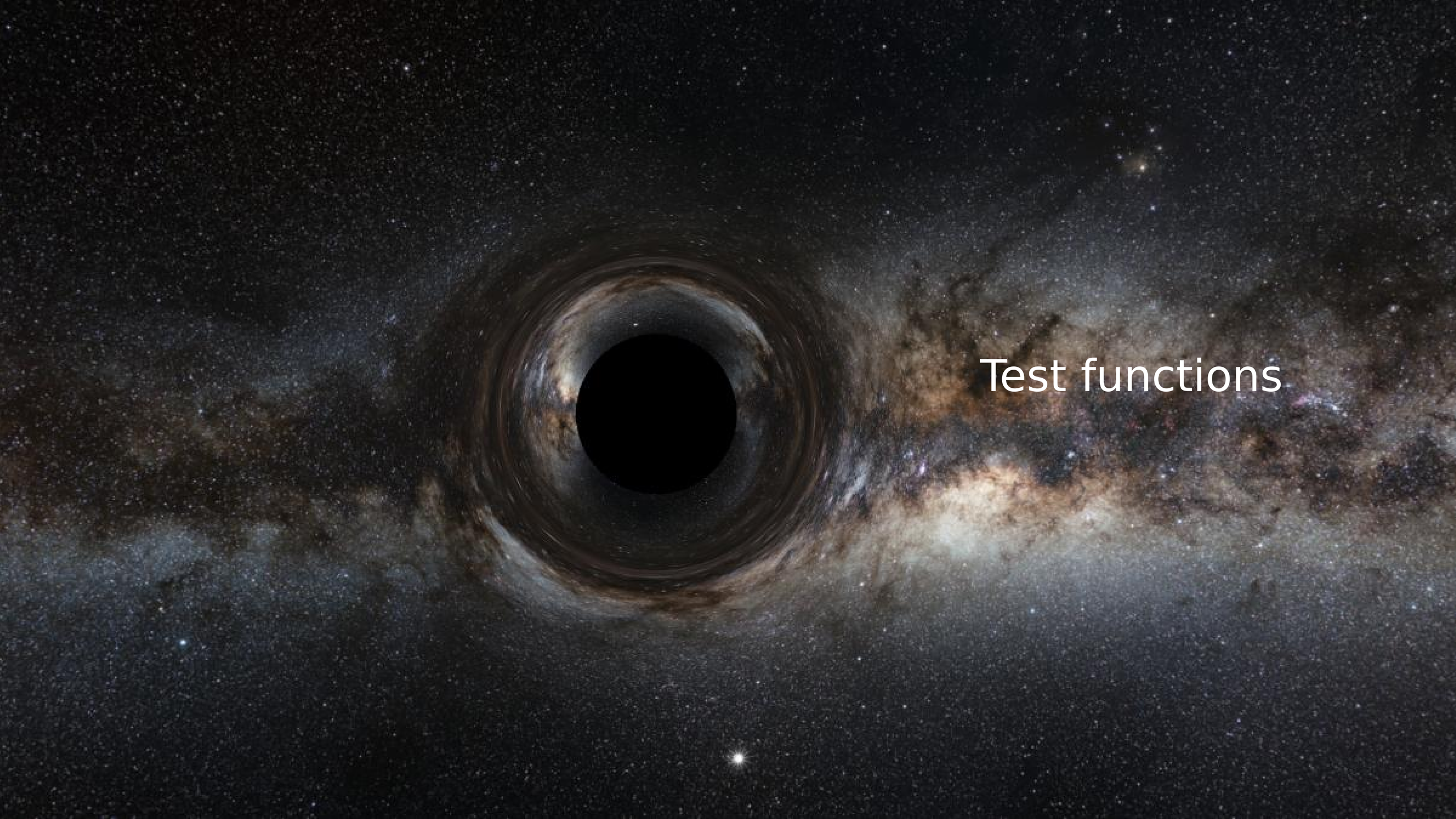
Simulated annealing

- Generating an initial temperature, and a random solution.
- Iterations until a satisfactory solution is obtained, or the temperature falls sufficiently.
- Generating a New Solution from the Modified Current.
- Replacement of the current solution by the new solution according to the measured energies:

```
newEnergy < energy ? true : Math.exp((energy - newEnergy) / temperature) >
    Math.random()
```

- Temperature decreases with each iteration `temp *= 1-coolingRate;` `coolingRate = 0.003`

Travelers salesman problem!



Test functions

Sphere

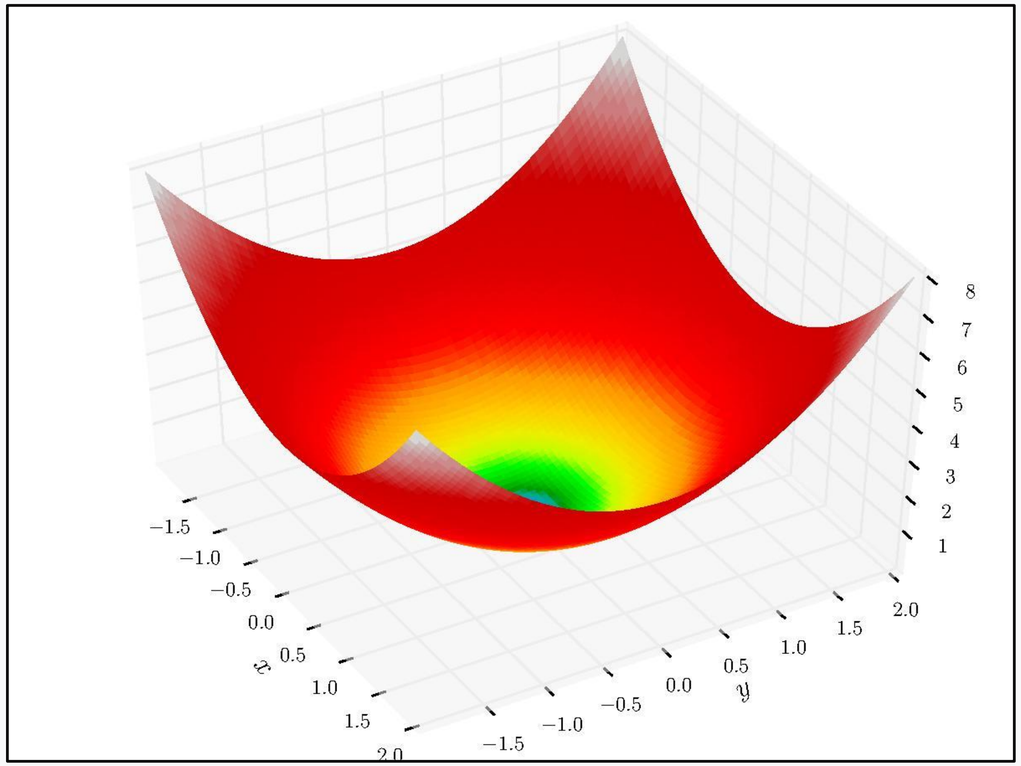
Formula:

$$f(\boldsymbol{x}) = \sum_{i=1}^{n} x_i^2$$

Minimum:

$$f(x_1, \ldots, x_n) = f(0, \ldots, 0) = 0$$

$$-\infty \leq x_i \leq \infty,$$
$$1 \leq i \leq n$$



Rosenbrock

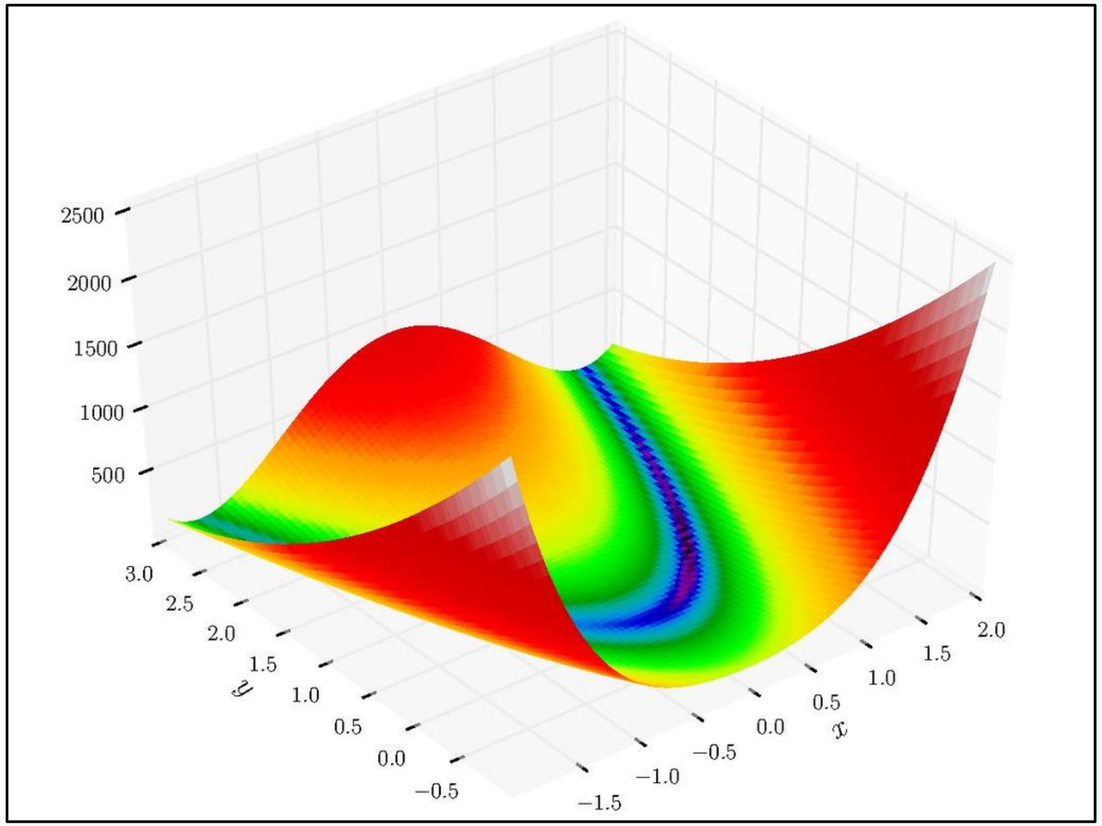
Formula:

$$f(\boldsymbol{x}) = \sum_{i=1}^{n-1} \left[100\left(x_{i+1} - x_i^2\right)^2 + \left(x_i - 1\right)^2 \right]$$

Minimum:

$$\text{Min} = \begin{cases} n = 2 & \rightarrow \quad f(1,1) = 0, \\ n = 3 & \rightarrow \quad f(1,1,1) = 0, \\ n > 3 & \rightarrow \quad f(\underbrace{1,\ldots,1}_{n \text{ times}}) = 0 \end{cases}$$

$$-\infty \leq x_i \leq \infty,$$
$$1 \leq i \leq n$$

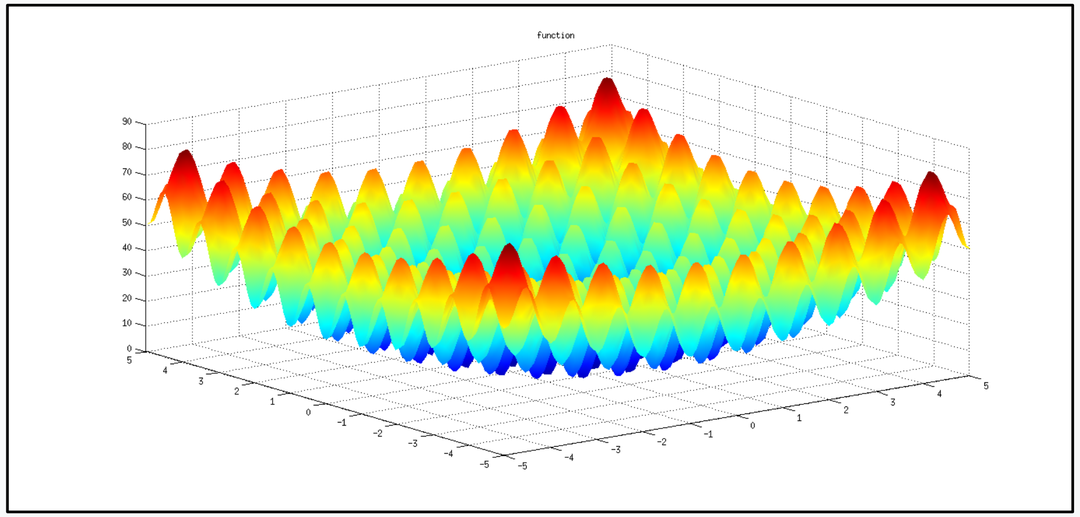


Rastrigin

Formula:

$$f(\mathbf{x}) = An + \sum_{i=1}^{n} \left[x_i^2 - A\cos(2\pi x_i)\right]$$

$$\text{where: } A = 10$$



Minimum:

$$f(0,0) = 0$$

$$-5.12 \leq x, y \leq 5.12$$

Bibliography

Clever Algorithms (Jason Brownlee):

https://users.info.uvt.ro/~dzaharie/ma2016/books/CleverAlgorithms_Brownlee_2011.pdf

The nature of code (Daniel Shiffman):

http://natureofcode.com/book/

Genetic Algorithms in Search, Optimization, and Machine Learning (David E. Goldberg):

https://www.amazon.com/Genetic-Algorithms-Optimization-Machine-Learning/dp/0201157675

Genetic.io

Questions?

Demo ! genetic.io/demo

Twitter ! @geneticio