

Fabiano Weimar dos Santos

# ALGORITMOS GENÉTICOS EM PYTHON



# Introdução

- Algoritmos Bio-inspirados
- Algoritmos Genéticos
  - Seleção
  - “Crossover”
  - Mutação
- Exemplo “Pythonico”

# Algoritmos Bio-inspirados

Se um dia desejarmos construir robôs que se comportam como seres autônomos, que tal programar esses robôs com algoritmos inspirados na **biologia**?

# Algoritmos Bio-inspirados

- Heurísticas Bio-Inspiradas:
  - Computação Evolutiva;
  - **Algoritmos Genéticos;**
  - Inteligência de Enxames (“Swarm Intelligence”);
  - Algoritmos Imunológicos;
  - Caos Determinístico;

# Algoritmos Genéticos

- Especialmente úteis quando as seguintes premissas se verificam:
  - Soluções para um problema podem ser encontradas, mas não é simples encontrar a melhor solução em um tempo aceitável";
  - Dada uma determinada solução é possível compará-la com outras soluções e dizer qual é a melhor;

# Algoritmos Genéticos

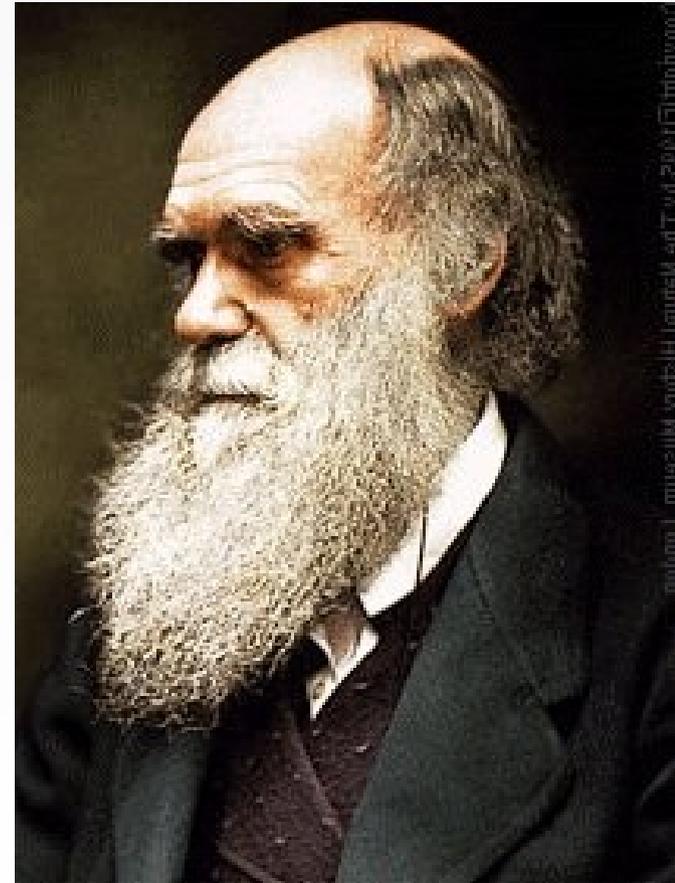
- É uma forma interessante de resolver problemas de otimização combinatória ou numérica.
  - Exemplo: o problema do Caixeiro Viajante (“Traveling Salesman Problem - TSP”).

# Algoritmos Genéticos

- Uma solução para o problema é um cromossomo;
- Operações de “crossover” e mutação nos cromossomos criam novas soluções;
- Mutações acontecem com uma determinada probabilidade, gerando soluções diferentes das “esperadas”;

# Algoritmos Genéticos

- Mecanismos de seleção fazem as novas soluções se reproduzirem ou serem extintas;



# Exemplo “Pythonico”

Achar o valor máximo da equação:

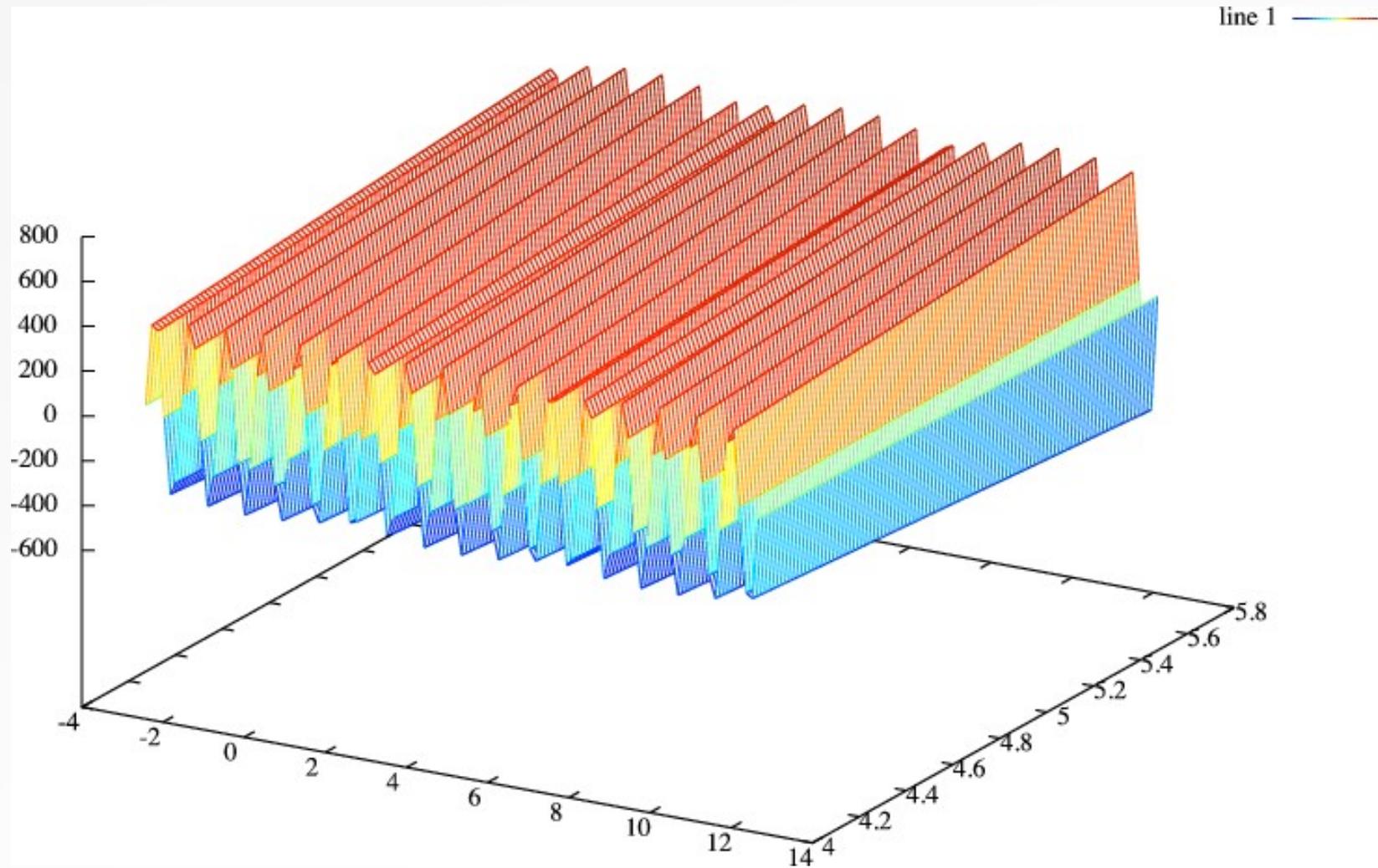
(todo mundo usa o TSP como exemplo, vamos usar algo diferente...)

$$f(x1, x2) = 21.5 + x1 \cdot \sin(4\pi x1) + x2 \cdot \sin(20\pi x2)$$

# Gráfico (octave)

```
x1 = linspace(-3.0, 12.1, 100);  
x2 = linspace(4.1, 5.8, 100);  
[xx1, xx2] = meshgrid(x1, x2);  
z1 = 21.5 + xx1 * sin( 4 * pi * xx1)';  
z2 = xx2 * sin(20 * pi * xx2)';  
z = z1 + z2;  
mesh(x1, x2, z);
```

# Gráfico (octave)



# Solução

Show me the code!

