

# Inspyred - Intro

Prof. Cesar M. V. Benitez  
2019

# Inspyred

- Criado por Aaron L. Garrett
- Baseado no livro “Evolutionary Computation: A Unified Approach” de Ken de Jong
- Objetivo do projeto:

biblioteca que separa “Computação de problema específico” e “de algoritmo específico” → Generalizar algoritmos para qualquer problema.

**Disponível no site:** <https://pythonhosted.org/inspyred/>

# Inspired

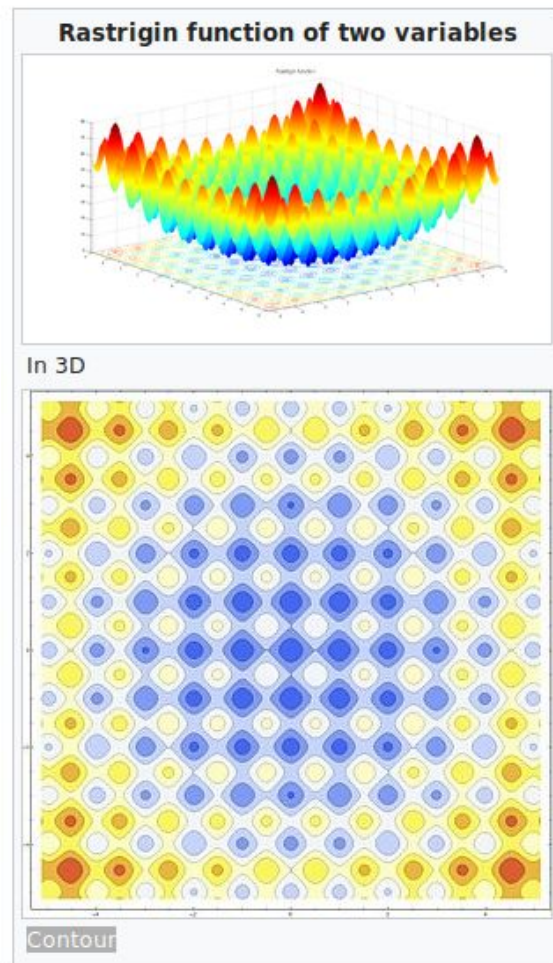
- Considera que algoritmos de Computação Evolucionária são compostos de:
  - Problem-specific components
    - A **generator** that defines how solutions are created
    - An **evaluator** that defines how fitness values are calculated for solutions
  - Algorithm-specific evolutionary operators
    - An **observer** that defines how the user can monitor the state of the evolution
    - A **terminator** that determines whether the evolution should end
    - A **selector** that determines which individuals should become parents
    - A **variator** that determines how offspring are created from existing individuals
    - A **replacer** that determines which individuals should survive into the next generation
    - A **migrator** that defines how solutions are transferred among different populations
    - An **archiver** that defines how existing solutions are stored outside of the current population

# Inspired - Exemplo

## - Minimização da Função Rastrigin

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

where  $A = 10$  and  $x_i \in [-5.12, 5.12]$ . It has a global minimum at  $\mathbf{x} = \mathbf{0}$  where  $f(\mathbf{x}) = 0$ .



# Inspyred - Exemplo

- O Gerador (generator): declara uma função que se comporta como iterator. Em outras palavras, declara uma função que pode ser usada em Loops (for).
- Neste exemplo: gera 10 numeros entre -5.12 e 5.12 (10 variaveis!)

```
from random import Random
from time import time
from math import cos
from math import pi
from inspyred import ec
from inspyred.ec import terminators
```

```
def generate_rastrigin(random, args):
    size = args.get('num_inputs', 10)
    return [random.uniform(-5.12, 5.12) for i in range(size)]
```

# Inspired - Exemplo

- O Evaluator: avalia todos os candidatos (solucoes candidatas)

```
def evaluate_rastrigin(candidates, args):  
    fitness = []  
    for cs in candidates:  
        fit = 10 * len(cs) + sum([((x - 1)**2 - 10 * cos(2 * pi * (x - 1))) for x in cs])  
        fitness.append(fit)  
    return fitness
```

Dica: Ver metodos append e extend.

Neste exemplo: fitness é uma lista com o fitness de cada candidato

# Inspyred - Exemplo

- Computação Evolucionária: o código fonte

Para maximização:  
Maximize = True

Para minimização:  
Maximize = False

```
rand = Random()
rand.seed(int(time()))
es = ec.ES(rand)
es.terminator = terminators.evaluation_termination
final_pop = es.evolve(generator=generate_rastrigin,
                      evaluator=evaluate_rastrigin,
                      pop_size=100,
                      maximize=False,
                      bounder=ec.Bounder(-5.12, 5.12),
                      max_evaluations=20000,
                      mutation_rate=0.25,
                      num_inputs=3)

# Sort and print the best individual, who will be at index 0.
final_pop.sort(reverse=True)
print(final_pop[0])
```

# Inspyred

- O código fonte

Completo:

Obs.: exemplo com

3 variáveis e

Evolution Strategy (ES)

```
#start_imports
from random import Random
from time import time
from math import cos
from math import pi
from inspyred import ec
from inspyred.ec import terminators
#end_imports

def generate_rastrigin(random, args):
    size = args.get('num_inputs', 10)
    return [random.uniform(-5.12, 5.12) for i in range(size)]

def evaluate_rastrigin(candidates, args):
    fitness = []
    for cs in candidates:
        fit = 10 * len(cs) + sum([((x - 1)**2 - 10 * cos(2 * pi * (x - 1))) for x in cs])
        fitness.append(fit)
    return fitness

#start_main
rand = Random()
rand.seed(int(time()))
es = ec.ES(rand)
es.terminator = terminators.evaluation_termination
final_pop = es.evolve(generator=generate_rastrigin,
                      evaluator=evaluate_rastrigin,
                      pop_size=100,
                      maximize=False,
                      bounder=ec.Bounder(-5.12, 5.12),
                      max_evaluations=20000,
                      mutation_rate=0.25,
                      num_input=3)
# Sort and print the best individual, who will be at index 0.
final_pop.sort(reverse=True)
print(final_pop[0])
#end_main
```

# Inspyred

- **Evolution Strategy (ES)** : “The default for an ES is to select the entire population, use each to produce a child via Gaussian mutation, and then use “plus” replacement. “
- **Como ejecutar o programa?**

**python rastrigin.py**

[1.0591544744884256, -0.06931594194249761, 2.014420132288964, 2.4373871664313254, 0.9287429640850096, 1.4371661116043493] : 3.83332404535

# Inspyred

## Algoritmo Genético (AG)

### Exemplo com

### Função Schwefel

```
from random import Random
from time import time
import inspyred

def main(prng=None, display=False):
    if prng is None:
        prng = Random()
        prng.seed(time())

    problem = inspyred.benchmarks.Binary(inspyred.benchmarks.Schwefel(2),
                                          dimension_bits=30)

    ea = inspyred.ec.GA(prng)
    ea.terminator = inspyred.ec.terminators.evaluation_termination
    final_pop = ea.evolve(generator=problem.generator,
                          evaluator=problem.evaluator,
                          pop_size=100,
                          maximize=problem.maximize,
                          boulder=problem.boulder,
                          max_evaluations=30000,
                          num_elites=1)

    if display:
        best = max(final_pop)
        print('Best Solution: \n{0}'.format(str(best)))
    return ea

if __name__ == '__main__':
    main(display=True)
```

# Inspyred

Ver no site do criador:

- Exemplos com outros algoritmos de Computação Evolucionária
- Como personalizar/customizar um algoritmo e função de fitness
- A modelagem dos problemas Knapsack e Traveling salesman

# Inspyred - Como instalar?

- Instale PIP (gerenciador de pacotes Python)

`Sudo apt install python-pip`

- Instale o Inspyred

`pip install inspyred`

- Instale o Matlib-plot, pylab e numpy