Algoritmo Genético com Python

"Fruto do acaso ou de um processo evolutivo?!"

~$ whoami

Ana Paula Mendes

Aspirante ao desenvolvimento web, pythonista, Co-fundadora da PyLadies Teresina, tenho um blog em construção chamado "Ana no terminal" =)

@anapauladsmendes /ananoterminal @ananoterminal

34.8 · National Council for Scientific Research, Lebanon

Is it possible to predict Lotto numbers using evolutionary algorithms?

Getting a lot of money from lottery tickets can create jubilation. For lottery players, it does not matter if the prize is just small. People nowadays are attempting to predict these numbers using different methods such statistical methods, heuristic and meta-heuristic methods. But, have they succeeded in their attempt? What was the percentage of success? Does it work for all type of Lottery machines? And what parameters are used in the prediction process?

Evolutionary Algorithms | Statistical Methods | Heuristics

[redacted]

Université Hassan II de Casablanca

8 months ago

Anyone who has the answer will play the lottery and I think he can not give it to you. it is a game of chance where the combinatorics are explosive. Each time the game is reset. there is no repetition so there is no law.

2 Recommendations

[redacted]

University M'Hamed Bougara of Boumerdes

8 months ago

I agree with Todor. No.


Happy Hunger Games!

Base biológica

"Fruto do acaso ou de um processo evolutivo?"

Achou que era fruto do acaso?

Base biológica:

"It is not the strongest of the species that survives, nor the most intelligent , but the one most responsive to change."

(Charles Darwin)

Base biológica:

- Inspirado em modelos biológicos:
  - Teoria da Evolução de Darwin (1859);
  - Genética de Mendel (1865);

Base biológica:

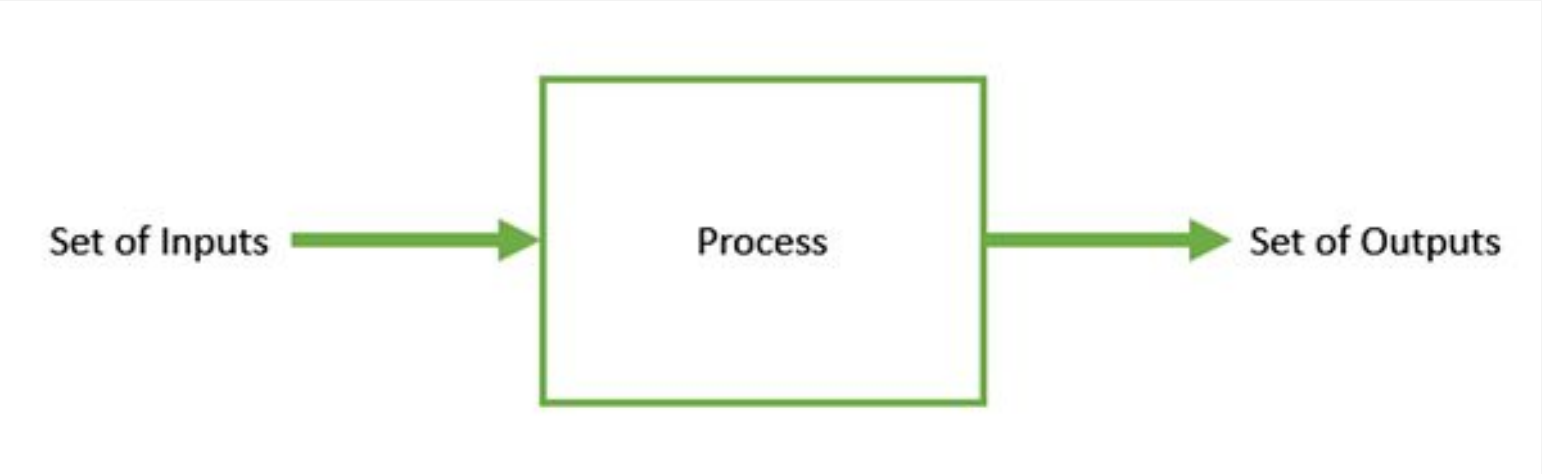
- Muitos termos foram emprestados da Biologia para os Algoritmos Genéticos;

O que são técnicas de otimização?

De onde vem? É de comer?


Isso é arte, rapá

O QUE SÃO TÉCNICAS DE OTIMIZAÇÃO?



O que são técnicas de otimização?

- Definição do termo "melhor" em otimização;
- Conjuntos das soluções;

O que são técnicas de otimização?

- Um problema de otimização pode ser representado da seguinte forma:
  - Dados: uma função $f : A \longrightarrow R$ de algum conjunto A de números reais;
  - Buscando: um elemento x0 em A tal que $f(x0) \leq f(x)$ para todo x em A ("minimização") ou tal que $f(x0) \geq f(x)$ para todo x em A ("maximização").

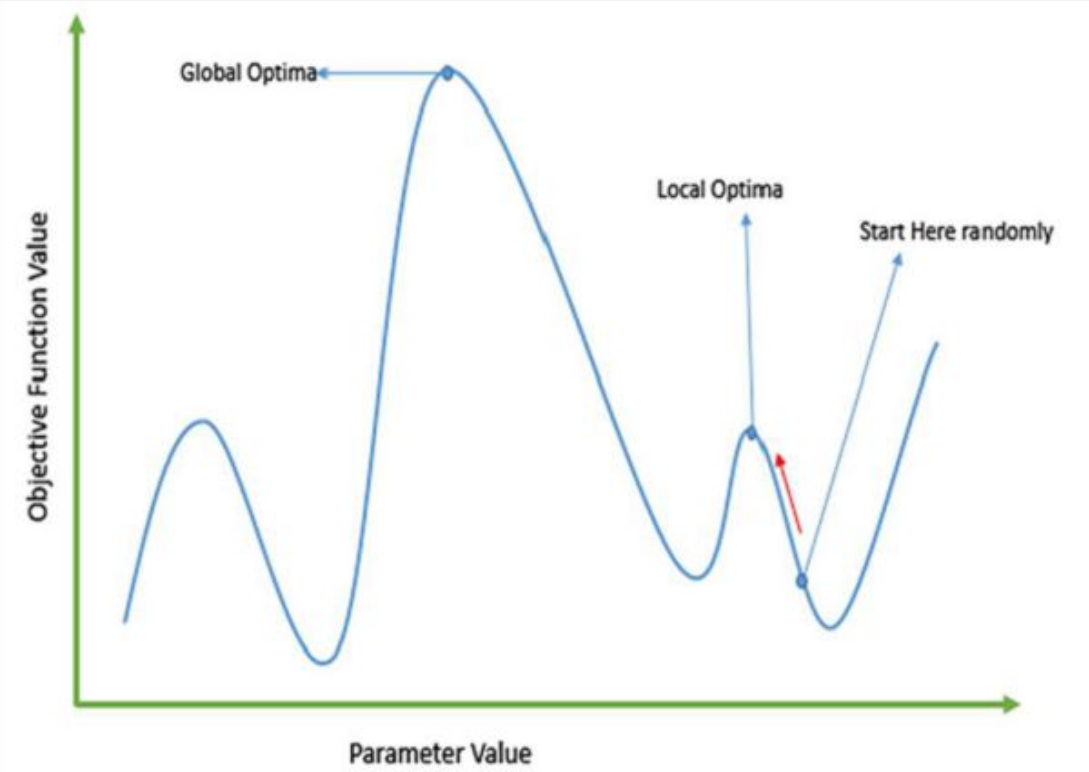
Algoritmo genético

Vamo programar uns dna agora é?!

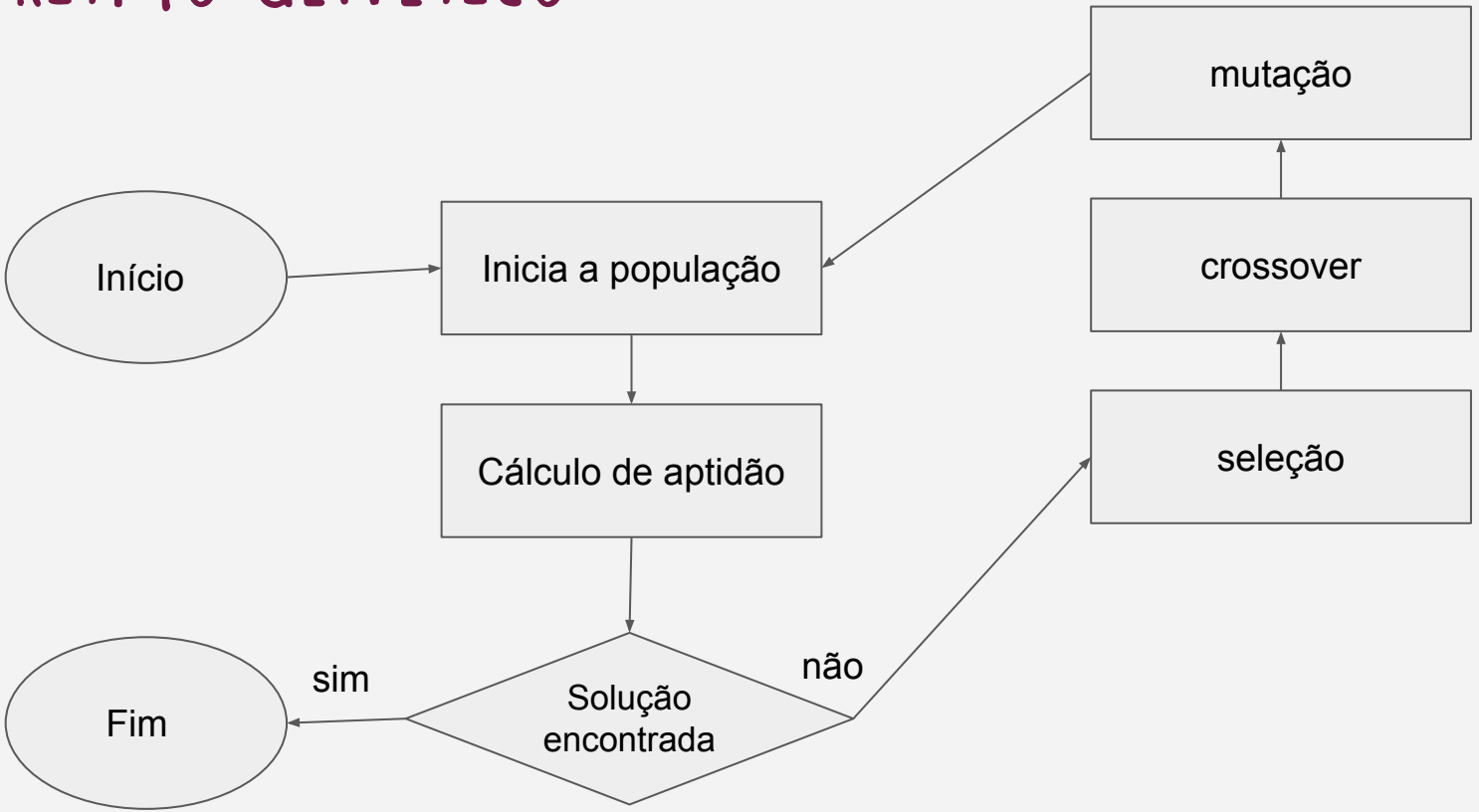
Introdução: Algoritmos genéticos

- Problemas NP;
- Falha de métodos baseados em gradientes;
- Obter uma boa solução rapidamente.

Falha de métodos baseados em gradientes



Algoritmo Genético



Algoritmos genéticos

- Inicialização da População:
  - Inicialização randômica;
  - Inicialização heurística.

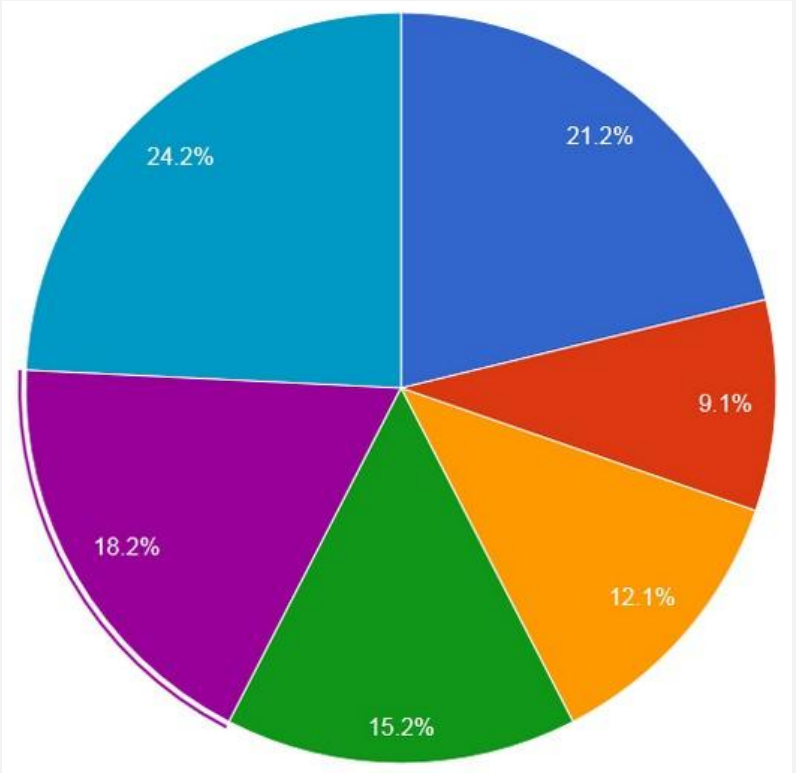
Algoritmos genéticos

- Cálculo de aptidão (Fitness):

  individuo[i] - modelo[i]

Algoritmos genéticos

- Seleção:
  - aleatória;
  - por torneio;
  - usando a roleta.

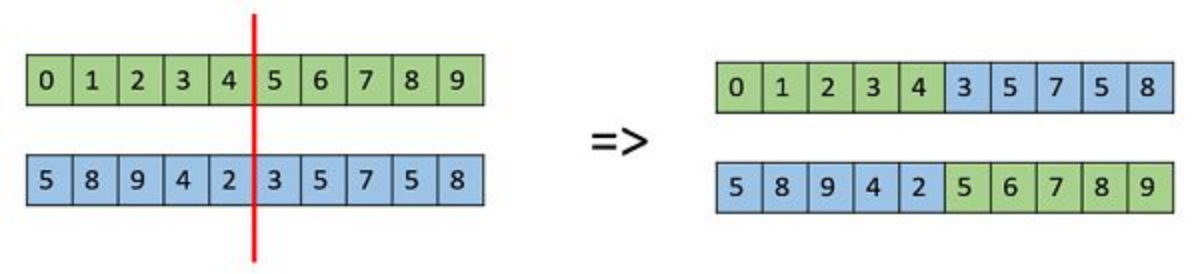


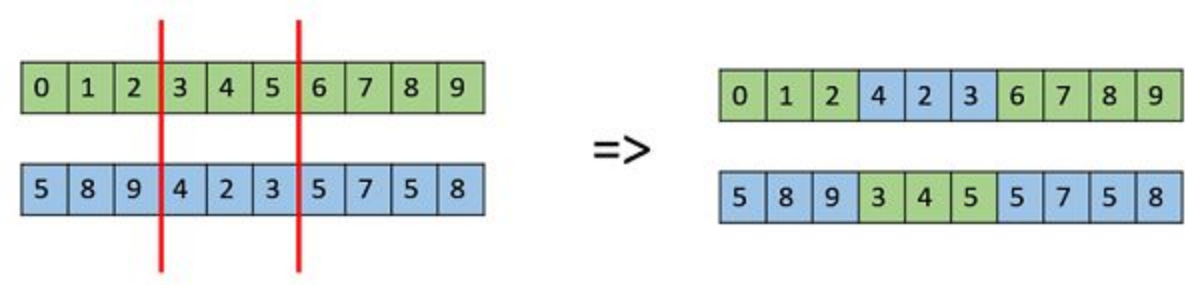
Algoritmos genéticos

- Cruzamento (Crossover):
  - Aleatório;
  - Single Arithmetic Crossover;
  - Simple Arithmetic Crossover;
  - Partially-Matched Crossover.

Cruzamento - Algoritmos genéticos

- Aleatório:





Cruzamento - Algoritmos genéticos

- Single Arithmetic Crossover:

$$\langle x_1, \ldots, x_k, \alpha \cdot y_k + (1-\alpha) \cdot x_k, \ldots, x_n \rangle$$

Pai 1={0.5, 1.0, 1.5, 2.0}, Pai 2={0.2, 0.7, 0.2, 0.7} a = 0.4 e k = 3

Filho 1={0.5, 1.0, (0.4)(0.2)+(0.6)(1.5), 2.0} = {0.5, 1.0, 0.98, 2.0}

Filho 2={0.2, 0.7, (0.4)(1.5)+(0.6)(0.2), 0.7} = {0.2, 0.7, 0.72, 0.7}

Cruzamento - Algoritmos genéticos

- Simple Arithmetic Crossover:

$$\langle x_1, \ldots, x_k, \alpha \cdot y_{k+1} + (1-\alpha) \cdot x_{k+1}, \ldots, \alpha \cdot y_n + (1-\alpha) \cdot x_n \rangle$$

Pai 1={0.5, 1.0, 1.5, 2.0}, Pai 2={0.2, 0.7, 0.2, 0.7} a = 0.4 e k = 3

Filho 1={0.5, 1.0, (0.4)(0.2)+(0.6)(1.5), (0.4)(0.7)+(0.6)(0.2)} = {0.5, 1.0, 0.98, 1.48}

Filho 2={0.2, 0.7, (0.4)(1.5)+(0.6)(0.2), (0.4)(2.0)+(0.6)(0.7} ={0.2, 0.7, 0.72, 1.22}

Cruzamento - Algoritmos genéticos

- Partially-Matched Crossover:

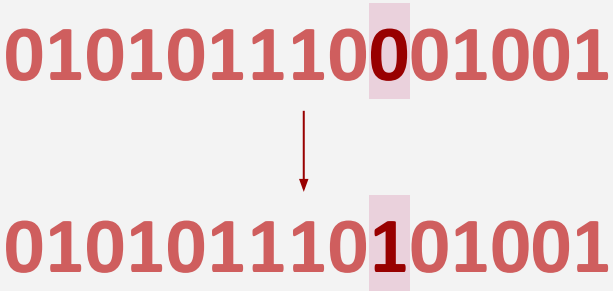
Pai 1 = 1 2 4 6 3 7 5 8 -> Substring 463
Pai 2 = 5 4 1 7 2 6 8 3 -> Substring 172

----

5 4 4 6 3 6 8 3
5 1 4 6 3 7 8 2 -> Números repetidos são substituídos

Mutação - Algoritmos genéticos

- Probabilidade de 1%:



Implementação com Python

Why so pythonic?

Criar a População:

```
def individuo(min, max):
    return[random.randint(min, max) for i in range(tam_individuo)]

def criarPopulacao():
    return[individuo(0,9) for i in range(tam_populacao)]
```

Função de Aptidão (Fitness)

```
def funcaoFitness(individuo):
    fitness = 0
    for i in range(len(individuo)):
        if(individuo[i] == modelo[i]):
            fitness += 1
    return fitness
```

Cruzamento (Crossover):

```
def selecaoEreproducao(populacao):
    pontuados = [(funcaoFitness(i), i) for i in populacao]
    pontuados = [i[1] for i in sorted(pontuados)]
    populacao = pontuados

    selecionados = pontuados[(len(pontuados) - pais):]

    for i in range(len(populacao) - pais):
        ponto = random.randint(1, tam_individuo - 1)
        pai = random.sample(selecionados, 2)

        populacao[i][:ponto] = pai[0][:ponto]
        populacao[i][ponto:] = pai[1][ponto:]

    return populacao
```

Mutação

```
def mutacao(populacao):
    for i in range(len(populacao) - pais):
        if(random.random() <= prob_mutacao):
            ponto = random.randint(0, tam_individuo - 1)
            novo_valor = random.randint(1, 9)

            while(novo_valor == populacao[i][ponto]):
                novo_valor = random.randint(1,9)

            populacao[i][ponto] = novo_valor

    return populacao
```

Bibliotecas de Algoritmo genético

- **Pyvolution:** Very modular and very extensible evolutionary algorithms framework, with complete documentation, Apache License 2.0;
- **deap:** Distributed Evolutionary Algorithms in Python, GNU Lesser GPL;
- **pySTEP:** Python Strongly Typed gEnetic Programming, MIT License;
- **Pyevolve:** Pyevolve was developed to be a complete genetic algorithm framework written in pure python;

Bibliotecas de Algoritmo genético

- PyRobot: Evolutionary Algorithms (GA + GP) Modules, Open Source;
- PonyGEA: small, one source file implementation of GE, with an interactive graphics demo application GNU GPL v3;
- INSPYRED: Biologically inspired computation encompasses a broad range of algorithms including evolutionary computation, swarm intelligence, and neural networks, GNU GPL v3;
- DRP: Directed Ruby Programming, Genetic Programming & Grammatical Evolution Library, GNU GPL;

Já acabou?

PESSOAS > TECNOLOGIA

Obrigada pela atenção, Galera!

@anapauladsmendes /ananoterminal @ananoterminal

Referências bibliográficas:

- Coppin, ben. Inteligência Artificial. Rio de janeiro: LTC, 2013.
- Artero, Almir Olivette. Inteligência Artificial: teoria e Prática. São Paulo: Editora Livraria da Física, 2009.


@anapauladsmendes


/ananoterminal


@ananoterminal