

Natural Selection meets Optimization: an Introduction to Genetic Algorithms

Arthur Nunes Torres e Mateus Hiro Nagata

LAMFO



Outline

1 Intuição

2 Formalismo

3 Extensões Naturais

- The Genetic Algorithm with Elitism

4 Genetic Algorithms meets real-life

- max MUSIC subject to Genetic Algorithm



O Teorema do Macaco Infinito

”Um macaco teclando aleatoriamente num teclado por um período infinito de tempo vai, quase-certamente, digitar qualquer texto, incluindo uma obra completa de William Shakespeare”.



Uma versão mais fraca do Teorema do Macaco Infinito -
Off topic



Figura: Fish beat Pokémon Sapphire



O Teorema do Macaco Infinito ligeiramente mais inteligente

$$\max \text{fitness}(x) \text{ s.a. tempo} = n < \infty$$

Solução: recursivamente selecionar o macaco que digita mais parecido com Shakespeare



Pipeline

- 1 Geração Natural
- 2 Seleção Natural
- 3 Continue até...



Contexto biológico

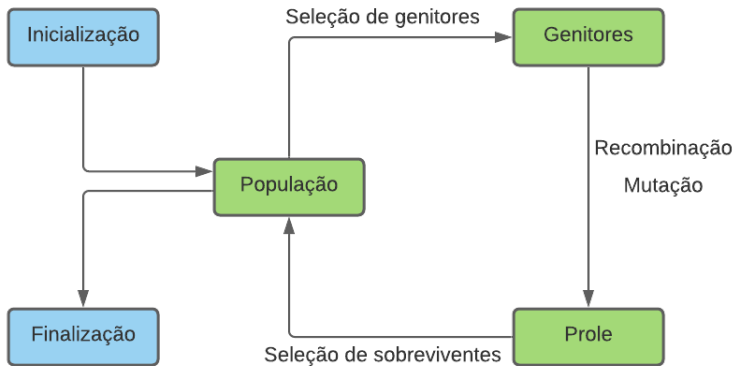
Os algoritmos evolutivos são algoritmos que buscam simular o processo natural de "tentativa e erro". Indivíduos são "testados" pelo ambiente, e os melhores adaptados se "perpetuam".

Os algoritmos evolutivos baseiam-se em 2 fenômenos biológicos:

- Evolução Darwiniana
- Genética



Contexto biológico



Outline

- 1 Intuição
- 2 **Formalismo**
- 3 Extensões Naturais
 - The Genetic Algorithm with Elitism
- 4 Genetic Algorithms meets real-life
 - max MUSIC subject to Genetic Algorithm



Estrutura de um algoritmo evolutivo genérico

COMEÇO

INICIALIZAR população com soluções aleatórias;

AVALIAR cada candidato;

Repetir até (CONDIÇÃO DE FINALIZAÇÃO é satisfeita)

1 SELECIONAR pais

2 RECOMBINAR pares de pais;

3 FAZER A MUTAÇÃO na prole resultante;

4 AVALIAR os novos candidatos;

5 SELECIONAR indivíduos para a próxima geração;



Componentes dos algoritmos evolutivos

- Representação (definição dos indivíduos)
- Função de avaliação (*fitness*)
- População
- Mecanismo de seleção de pais
- Operadores de variação (Recombinação e mutação)
- Mecanismo de seleção de sobreviventes

Representação dos indivíduos

A representação fará o mapeamento do "genótipo" para o "fenótipo".

Os indivíduos podem ser representados como:

- Bit-strings

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 1 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|

- Real-valued strings $\langle x_1, \dots, x_k \rangle, x_i \in \mathbb{R}$

- Árvores

Modelos de gestão de população

O principal modelo de gestão da população chama-se **modelo geracional**, no qual toda a população da geração n é substituída pela de $n + 1$.

Ou seja, para uma população de tamanho μ , que gera uma prole de tamanho λ , temos que $\mu = \lambda$.



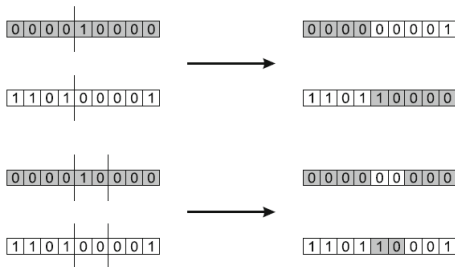
Mecanismo de seleção de pais

A cada um dos indivíduos da população, é atribuída uma probabilidade de ser selecionado. Ela pode ser proporcional ao valor de *fitness*, ao *rank* ou ser uniforme.



Operadores de variação - Recombinação

Tendo selecionado os pais, um ponto aleatório r é selecionado no espaço $[1, l - 1]$.



Operadores de variação - Mutação

Um ou mais indivíduos podem sofrer mutação. No caso de Bit-strings, alguns bits tem probabilidade p_m de mudar.



Outline

- 1 Intuição
- 2 Formalismo
- 3 Extensões Naturais**
 - The Genetic Algorithm with Elitism
- 4 Genetic Algorithms meets real-life
 - max MUSIC subject to Genetic Algorithm



The Genetic Algorithm with Elitism

```
def GA_elitism(popsize, n):  
    popsize = popsize    #desired population size  
    n = n #desired number of elite individuals  
    P = []  
    for i in range(popsize):  
        P.append(new_random_individual)  
    Best = None  
    while Best != ideal_solution or run_out_of_time == False:  
        for j in P:  
            AssessFitness(Pi)  
            if Best = None or Fitness(Pi) > Fitness(Best):  
                Best = j
```



The Genetic Algorithm with Elitism

```
Q =  
for k in range((popsize-n)/2):  
    P_a = SelectWithoutReplacement(P)  
    P_b = SelectWithoutReplacement(P)  
    C_a, C_b = Crossover(Copy(P_a), Copy(P_b))  
    Q.append(Mutate(C_a), (C_b))  
P = Q  
return Best
```

Elitismo

- Ant Colony Optimization algorithms
- Scatter Search
- Particle Swarm Optimization



Pequenas Variações

- 1 Steady-State Genetic Algorithm: filhos só entram na nova geração se tiverem maior fitness que alguém da população
- 2 Differential Evolution: filhos competem com os pais para inclusão. Além disso, filhos nascem de $\vec{a} + \alpha(\vec{b} - \vec{c})$
- 3 Hybrid Optimization Algorithm: juntar algoritmo evolutivo com escalada
- 4 Lamarckian Algorithms: algoritmo global com busca local
- 5 Particle Swarm Optimization (PSO): mutação direcionada para a média ponderada da melhor direção dada pelos informantes, pelo enxame todo e pelo indivíduo



Outline

- 1 Intuição
- 2 Formalismo
- 3 Extensões Naturais
 - The Genetic Algorithm with Elitism
- 4 Genetic Algorithms meets real-life
 - max MUSIC subject to Genetic Algorithm



Coding a Genetic Algorithm

- generate_genome()
- generate_population()
- single_point_crossover()
- mutation()
- population_fitness()
- selection_pair()
- generate_weighted_distribution()
- sort_population()
- genome_to_string()
- run_evolution()



Pipeline

Video from Kie Codes

Code from Kie Codes

- 1 Gerar n indivíduos (melodias)
- 2 Nota $\in \{0,1,2,3,4,5\}$ para cada indivíduo (meloda)
- 3 Manter as 2 melhores, combinar com as outras e fazer mutações aleatórias.
- 4 Gerar a próxima geração e repetir até estar satisfeito

The Genome

- 1 8 bars
- 2 1 bar = 4 beats
- 3 1 beat = 2 notes
- 4 1 note = 16 different pitches
- 5 16 pitches = $2^4 = 4$ bits
- 6 BPM 120

$$8bars \times 4beats \times 2notes \times 4bits = 256bits$$



Music

- A minor
- 16 bars
- Me: drum, bass, piano, pad
- Genetic Algorithm: arpeggiator, violin, pluck



max MUSIC subject to Genetic Algorithm

Software - Studio One 4

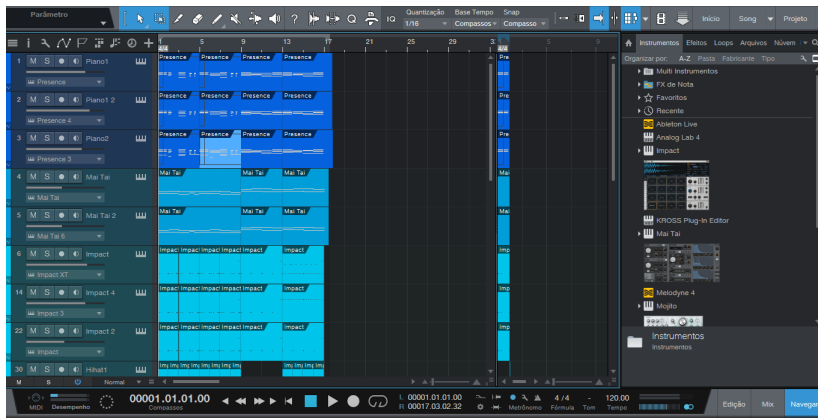


Figura: Caption

max MUSIC subject to Genetic Algorithm

Software - Studio One 4

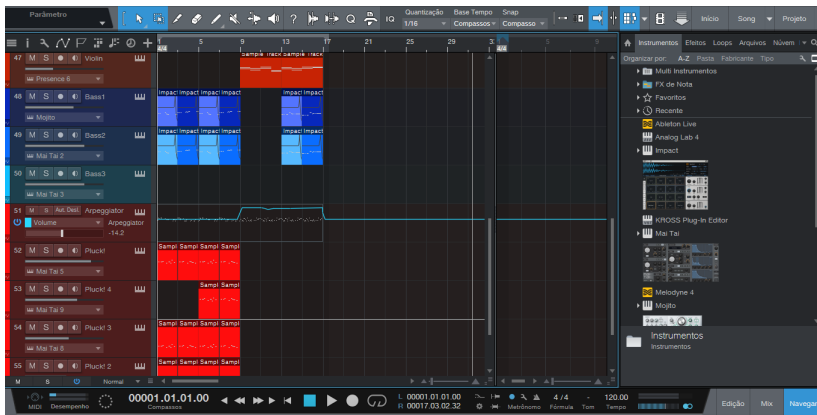


Figura: Caption

Natural Selection meets Optimization: an Introduction to Genetic Algorithms

Arthur Nunes Torres e Mateus Hiro Nagata

LAMFO



Pósfacio

- Instituições ficam presas em ótimos locais?
- Competição é o melhor jeito de passar os "genes" na economia?
- E se o terreno for acidentado?
- Genetic Algorithm pode ser um preditor? Ou é só um benchmark?



Referências

- 1 Video from Kie Codes
- 2 Code from Kie Codes
- 3 Brabazon, A., O'Neill, M., McGarraghy, S. (2015). Introduction to evolutionary computing. In Natural Computing Algorithms (pp. 17-20). Springer, Berlin, Heidelberg.
- 4 Luke, S. (2013). Essentials of Metaheuristics. Lulu, 2013. Metaheuristics in large-scale global continues optimization: A survey, available at: <http://cs.gmu.edu/sean/book/metaheuristics>.