

# Algoritmos Genéticos

---

Diego Addan

Unibrasil 2019

# Características

**Geração:** Criação das soluções.

**População:** Conjunto de indivíduos que compõe uma geração.

**Função de Fitness:** Função de avaliação que mede aptidão/ambiente.

# Características

**Seleção:** Escolha de indivíduos (apresentam maior score).

**Crossover:** Aplicação de operadores genéticos de mutação. Definido por indivíduos com melhor score e suas soluções.

**Critério de parada:** Obtenção de uma solução satisfatória por aproximação ou tempo computacional estabelecido.

# Tipos de representação

Cada indivíduo é considerado um cromossomo, codificado como um vetor de caracteres de tamanho fixo.

**Representação Binária:** Quando a solução não é dessa natureza, é feita a conversão transformando possíveis soluções em sequências {0,1}.

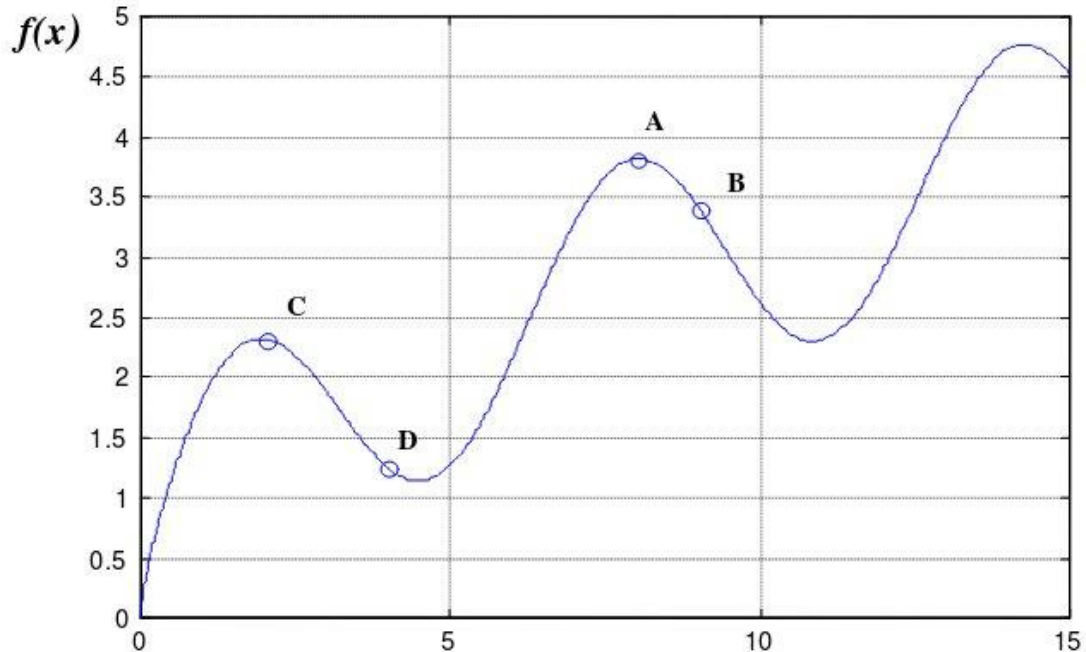
Cromossoma A	1	5	3	2	6	4	7	9	8
Cromossoma B	8	5	6	7	2	3	1	4	9

**Representação em ponto flutuante:** Cada cromossomo é formado pela concatenação de números reais. Apresenta vantagens em relação a representação binária.

Cromossoma A	1.2324	5.3243	0.4556	2.3293	2.4545
--------------	--------	--------	--------	--------	--------

# Tipos de representação

**Representação Binária:** Maximização da função  $f(x) = \text{sen}(x) + \sqrt{x}$



- $x$  assume valores no intervalo  $[0,15]$ .
- Cada indivíduo (solução), deverá possuir 4 bits.

# Tipos de representação

Neste problema, a própria função  $f(x)$  pode ser utilizada como fitness.

- **Exemplo de uma população de indivíduos com codificação binária**

Candidato à solução	Codificação (genótipo)	$x$ (fenótipo)	Fitness = $f(x)$
A	1 0 0 0	8	3,8178
B	1 0 0 1	9	3,4121
C	0 0 1 0	2	2,3235
D	0 1 0 0	4	1,2432

O melhor indivíduo neste caso é 1110, correspondente a  $x = 14$  e  $f(x)=4,7323$

# Tipos de representação

**Representação em ponto flutuante:** Cromossomos são representados de tamanho menor, reduzindo o processamento computacional. A decodificação se torna desnecessária.

Similaridade com aplicações de métodos e soluções numéricas. Não permite, no entanto conceitos de blocos construtivos para demonstração de convergência no processo evolutivo.

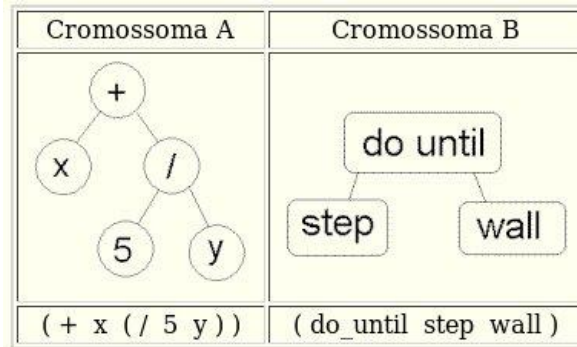
Problema de minimização de perdas.

# Tipos de representação

**Representação em valores:** É necessário desenvolver métodos de crossover e mutação específicos, além da aptidão.

Cromossoma A	1.2324 5.3243 0.4556 2.3293 2.4545
Cromossoma B	ABDJEIFJDHDIERJFDLDFLFEGT
Cromossoma C	(atrás), (atrás), (direita), (frente), (esquerda)

**Representação em árvore:** Cromossomos são nós de objetos como funções, comandos, estruturas.



# Tipos de representação

## **Tamanho da população**

O tamanho da população inicial e da população nas subseqüentes gerações é estabelecida empiricamente.

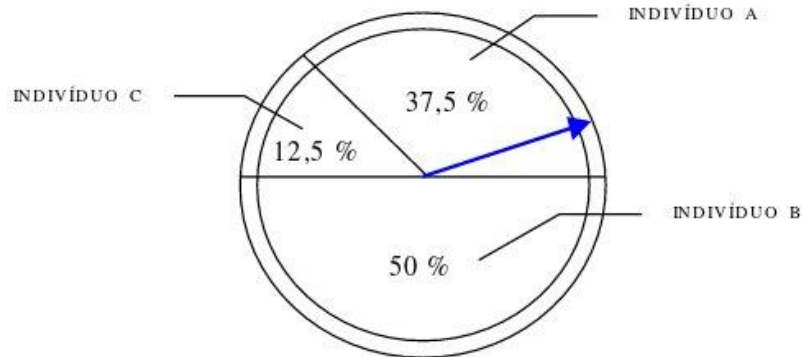
## **Mecanismos de Seleção**

Roulette Wheel (Goldberg, 1989), ou Seleção por torneio.

# Tipos de representação

**Seleção por Torneio:** Cada indivíduo é associado a porção de uma roleta, proporcional ao seu valor de fitness.

INDIVÍDUO	FITNESS	PERCENTAGEM DO TOTAL
A	75	37,5
B	100	50,0
C	25	12,5
TOTAL	200	100,0



# Tipos de representação

## **Mecanismos de Seleção**

Indivíduos mais adaptados tem maior chance de ser escolhido, mas respeita-se a aleatoriedade com o objetivo de manter a diversidade da população.

Pode ser definido empiricamente por critério de solução, como no caso OneMax (maior valor no somatório)

# Tipos de representação

## **Função Fitness**

Em termos naturais/genéticos, representa adaptabilidade.

Em termos computacionais indica o quão bem uma solução é capaz de resolver o problema.

Exemplo: Aproximação em métodos aplicados.

# Tipos de representação

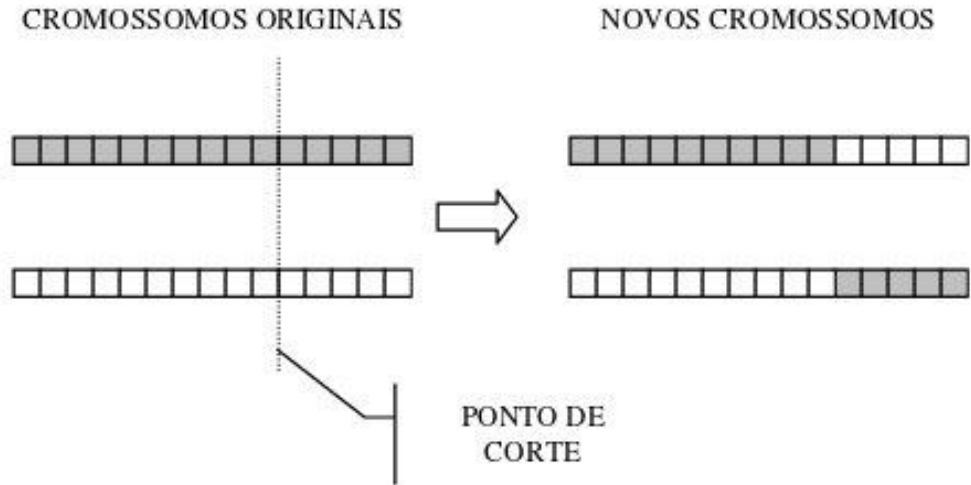
## Operadores Genéticos

A codificação do indivíduo determina quais operadores genéticos serão aplicados.

O operador de crossover pode ser **simples**, **multiponto**, **uniforme ordenado**, **aritmético**, ou uma combinação deles.

Mutação binária

Mutação float.



# Exemplo de aplicação de AGs

## Problema Knapsack (Mochila)

Vários itens que gostaria de levar na mochila.

Cada item tem um peso e um benefício.

Há uma capacidade limite de peso.

Deve-se carregar itens com o máximo valor total sem superar o limite de peso.

# Exemplo de aplicação de AGs

## Problema Knapsack

Problema de otimização combinatória (NP-Completo).

Resolvido por diversos algoritmos e métodos numéricos.

Aplicações:

- Corte e empacotamento
- Carregamento de recursos (digitais e alocação em disco)

# Exemplo de aplicação de AGs

## Problema Knapsack

### Exemplo

- item :	1	2	3	4	5	6	7
- Benefício	5	8	3	2	7	9	4
- Peso	7	8	4	10	4	6	4

Mochila suporta, no máximo, 22 quilos.

Adicione itens de modo a ter o máximo benefício,

# Exemplo de aplicação de AGs

## Problema Knapsack

- Codificação: 0 = não existe, 1 = existe na mochila

Cromossomo: 1010110

Item	1	2	3	4	5	6	7
Cromossomo	1	0	1	0	1	1	0
Existe?	sim	não	sim	não	sim	sim	não

→ Itens pegos: 1, 3, 5, 6

- Geração aleatória de população com  $n$  cromossomos:
  - a) 0101010
  - b) 1100100
  - c) 0100011

# Exemplo de aplicação de AGs

## Problema Knapsack: Fitness e seleção

a) 0101010: Benefício = 19, Peso = 24 ✗

Item	1	2	3	4	5	6	7
Cromossomo	0	1	0	1	0	1	0
Benefício	5	8	3	2	7	9	4
Peso	7	8	4	10	4	6	4

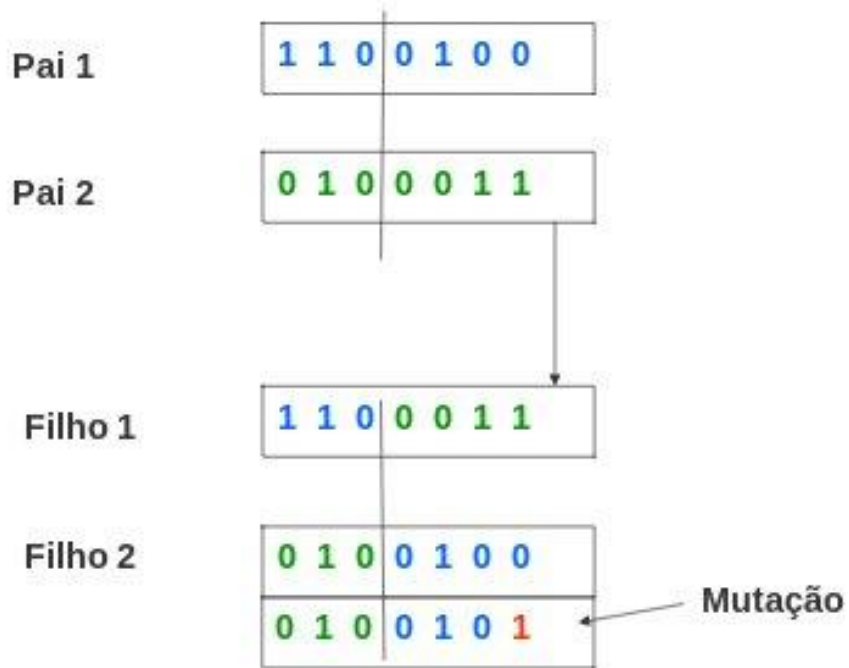
b) 1100100: Benefício = 20, Peso = 19 ✓

c) 0100011: Benefício = 21, Peso = 18 ✓

**Serão selecionados os cromossomos b & c.**

# Exemplo de aplicação de AGs

## Problema Knapsack: Crossover e mutação



# Exemplo de aplicação de AGs

## Problema Knapsack

Aceitação, substituição e teste:

- Definição de nova geração em uma nova população.
- Uso de nova população gerada para uma próxima rodada do algoritmo.
- Se a condição final é satisfeita, então finaliza a execução.
  - **Condições de parada**
  - Número de gerações
  - Melhoramento da melhor solução.
- Caso contrário, retorne ao passo de fitness.