

# Fundamentos de Algoritmos Genéticos



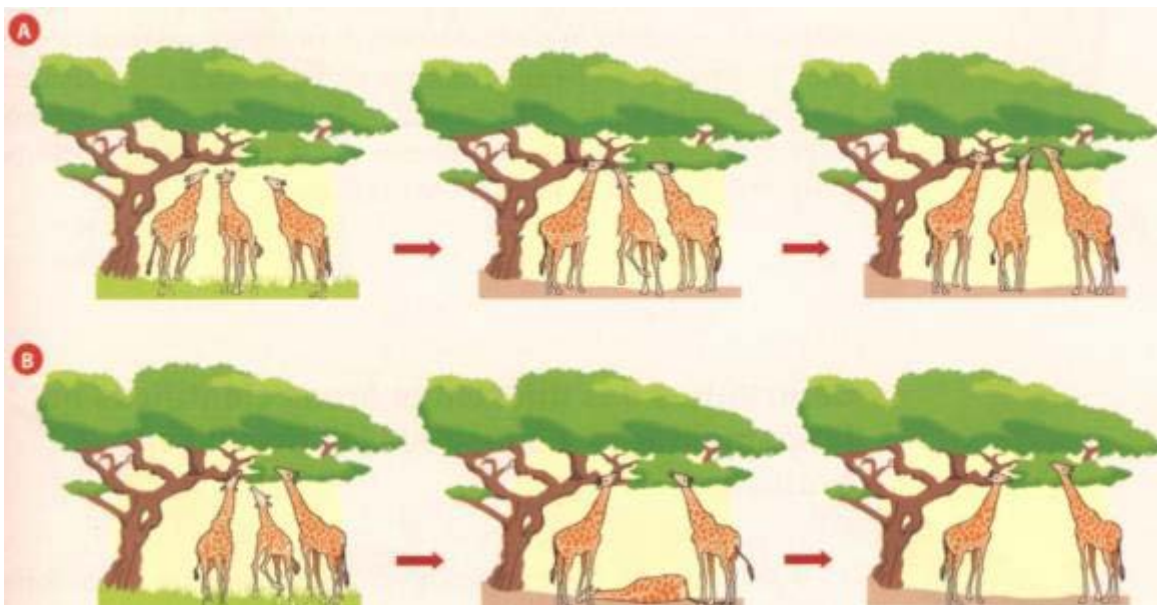
## Algoritmos Genéticos – Inspiração

### A Seleção Natural é um processo de Otimização Biológica.

No mundo natural há uma tremenda diversidade de organismos.

O grau de complexidade nos organismos é alto.

Os organismos de hoje são resultado de muitas iterações de um grande algoritmo de otimização adaptativo/evolutivo, que, através de saltos evolutivos, modificou morfológicamente e fisiologicamente indivíduos de forma a adequá-los ao ambiente (vide "*Evolutionary leap from fins to legs was surprisingly simple*" em <https://www.sciencedaily.com/releases/2016/03/160308090800.htm>)



Salto evolutivo modificou adaptativamente o tamanho do pescoço de girafas ao longo das gerações de modo aos indivíduos serem aptos a alcançar o alimento nas árvores.

Girafas com pescoço curto não alcançam o alimento nas árvores e morrem de modo que, em função do cruzamento entre indivíduos ao longo das gerações, os genes que se tornaram dominantes são os das girafas com pescoço

## Que mecanismo dita as regras de cada *evolutionary leap*?

Evolutionary biology has fascinated scientists since Charles Darwin who cornered the concept of natural selection in the 19th century. Accordingly, organisms better adapted to their environment tend to survive and produce more offspring; in other terms, randomly occurring mutations that render the organism more fit to survival will be carried on and be transmitted to the offspring. Nearly a century later, science has seen the discovery of quantum mechanics, the branch of mechanics that deals with subatomic particles. Along with it, came the theory of quantum evolution whereby **quantum effects can bias the process of mutation towards providing an advantage for organism survival**. This is consistent with looking at the biological system as being a product of chemical-physical reactions, such that chemical structures arrange according to physical laws to form a replicative material referred to as the DNA. In this report, we attempt to reconcile both theories, trying to demonstrate that they complement each other, hoping to fill the gaps in our understandings of the versatility of the mutational status of the DNA as an essential mechanism of life compatibility.

**Fonte:** Darwinian Evolution and Quantum Evolution are Complementary: A Perspective

<https://www.longdom.org/open-access/darwinian-evolution-and-quantum-evolution-are-complementary-a-perspective-DOI-2161-1041-1000181.pdf>

## Otimização Biológica Evolutiva / Algoritmo de Otimização Adaptativa

A **função de custo** de um algoritmo de otimização adaptativo mede o quão distante está o algoritmo da solução ótima. Por exemplo, em uma população de girafas o erro quadrático entre a altura da cabeça de um indivíduo girafa e a altura do alimento nas árvores mede o quão distante está o indivíduo da solução ótima (alcançar o alimento na árvore) no processo de a população de girafas se adaptar às condições ambientais.

No grande algoritmo de otimização adaptativa da mãe-natureza isto pode ser visto como uma medida do erro instantâneo em uma iteração do processo adaptativo que ocorre durante milênios, de salto evolutivo em salto evolutivo, ao longo das gerações de girafas.

Mas também pode ser visto como uma medida da inaptidão do indivíduo no processo evolutivo de geração em geração em que somente os mais aptos são selecionados a sobreviver.

Uma função que varie inversamente com a função de custo mede o nível de aptidão do indivíduo para implementar uma solução ótima do processo evolutivo – denominada de **função de fitness**.

Os indivíduos cuja avaliação resulta nos maiores valores de *fitness* são os que tem melhores condições de interagir com as restrições do ambiente e sobreviver.

O processo de evolução é o algoritmo que seleciona (cruzamento entre os indivíduos mais aptos) e ajusta (mutação – “tutor” quântico?) quais características produzem um organismo adequado para sobrevivência.

## Evolução como Inspiração

Problemas computacionais requerem:	A evolução permite:
<p><b>Busca ao longo de um número muito grande de possíveis soluções.</b></p>	<ul style="list-style-type: none"> <li>• Número de possibilidades: conjunto de sequências genéticas.</li> <li>• Soluções desejadas: organismos altamente ajustados (adequados), perfeitamente capazes de sobreviver e de se reproduzir em seus ambientes.</li> </ul>
<p><b>Capacidade de inovar (extrapolar soluções).</b></p>	<ul style="list-style-type: none"> <li>• A evolução permite projetar novas soluções para problemas complexos (sistema imunológico dos mamíferos face à constante invasão do corpo por germes).</li> </ul>
<p><b>Adaptatividade .</b> (continuar a desempenhar adequadamente em ambientes não-estacionários)</p>	<ul style="list-style-type: none"> <li>• A adequação de um organismo biológico depende de vários fatores: como resiste às características físicas do ambiente, como compete ou coopera com outros organismos.</li> <li>• O critério de adequação muda continuamente, à medida que a criatura evolui.</li> </ul>
<p><b>Soluções complexas, difíceis de programar.</b> (inteligência artificial -&gt; regras muito simples permitem a emergência de comportamentos complexos (como inteligência) através da aplicação fortemente paralela destas regras e da interação entre elas)</p>	<ul style="list-style-type: none"> <li>• A evolução não se ocupa de uma espécie a cada vez, de forma fortemente paralela testa e muda milhões de espécies em paralelo.</li> </ul>

## Algoritmos Genéticos - Definição

*Algoritmos Genéticos foram criados e formalmente descritos por John Holland, nos anos 60.*

*“Algoritmo evolucionário que modela processos biológicos para otimizar funções de custo altamente complexas.”*

Um algoritmo genético permite a **evolução** de uma **população** composta de muitos **indivíduos**, sobre **regras de seleção especificadas**, para um estado que **maximize** o **fitness** da população final, ou seja, maximize a adaptação da população à tarefa de interesse (o que equivale a um processo de minimização de função de custo).

## Um Algoritmo Genético é um modelo que:

Trata uma **população de soluções** de um determinado problema, aplicando determinadas **operações** aos **membros** da população (indivíduos), de forma que a população resultante seja composta por **indivíduos mais aptos** do que a população inicial, ou até que se selecione apenas **o indivíduo mais apto**.

A maior parte dos métodos chamados GAs possuem, no mínimo, os seguintes elementos em comum:

- População de indivíduos caracterizados por cromossomos
- Seleção de acordo com o *fitness* do indivíduo.
- Cruzamentos para produzir uma nova geração (prole)
- Mutação aleatória da nova prole.

## Um GA pode ser resumido como:

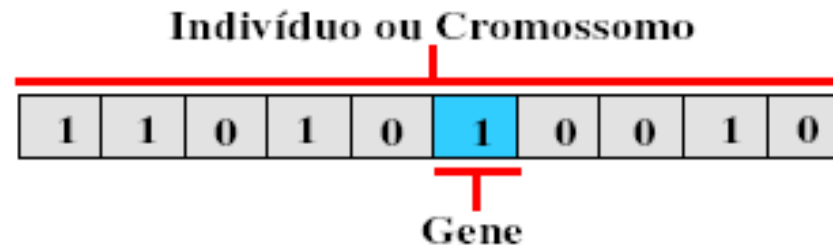
<p>(1) Gerar um conjunto de soluções candidatas, que constituem a população inicial;</p>	<p>Um GA não calcula uma única solução p/ o problema que está sendo tratado. <b>Cada um dos indivíduos</b> <math>\in</math> à população representa uma <b>solução completa</b> do problema que se deseja tratar;</p>
<p>(2) Avaliar a adequação das soluções candidatas de acordo com algum critério de <i>fitness</i>;</p>	<p>A qualidade de cada possível solução é avaliada através de uma função <b>de adequação do indivíduo à tarefa</b> (<i>fitness function</i>: função que avalia a <b>aptidão</b> do indivíduo na execução da tarefa);</p>
<p>(3) Decidir, de acordo com a avaliação do fitness, quais indivíduos serão mantidos e quais serão descartados;</p>	<p>A avaliação é feita não apenas de forma <b>absoluta</b> (<i>evaluation</i>), mas também de forma <b>relativa</b> aos demais indivíduos da população (<i>fitness</i>);</p>
<p>(4) Produzir variantes da população (novos indivíduos) através do uso de alguns tipos de operadores sobre a população de candidatos sobreviventes.</p>	<p><b>São alocadas propriedades seletivas e reprodutivas</b>, de tal forma que os indivíduos que representem melhores soluções p/ o problema alvo tenham mais chances de persistir ou de se reproduzir no conjunto de soluções (as melhores soluções são encorajadas).</p>



## População

As soluções p/ o problema alvo são codificadas em estruturas que possam ser armazenadas e operadas computacionalmente.

As estruturas codificadas são chamadas genomas ou cromossomos, e são sequências de 0s e 1s (caso binário).



Um algoritmo genético:

- Cria uma **população de cromossomos** tipicamente aleatória;
- Aplica operações de ***crossover, mutation & selection*** para gerar novas soluções, no espaço de busca.

## Espaços de Busca - Definição

- O termo “*search space*” se refere a alguma **coleção de soluções candidatas** para o problema alvo e a uma **noção de distância entre soluções candidatas**.
- GAs assumem que indivíduos pais de alto *fitness*, localizados em diferentes regiões no espaço de busca podem ser combinadas por meio de cruzamento para produzir soluções candidatas (indivíduos filhos) de alto *fitness*.

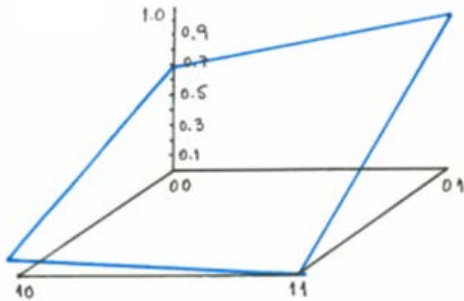
## Cenários de Fitness

Definido (em 1931) pelo biólogo [S.Wright](#) no contexto de genética populacional.

É uma representação do espaço de todos os possíveis padrões binários de genes, associados a seus particulares *fitness*.

Supondo que:

- Cada gene é uma palavra de  $l$  bits;
- A distância entre 2 genes é medida pela distância de *Hamming* (que equivale ao número de posições nas quais bits correspondentes diferem);
- A cada gene é atribuído um valor de *fitness*.
- Um cenário de *fitness* pode ser descrito como um gráfico  $(l+1)$  dimensional no qual cada um dos  $2^l$  genes é um ponto em localizado em um espaço de  $l$  dimensões e seu *fitness* é plotado ao longo do eixo do espaço de ordem  $(l+1)$ .



A figura apresenta um simples cenário (ou paisagem) de fitness para  $l=2$ .

$$f(00)=0.7; f(01)=1.0; f(10)=0.1; f(11)=0$$

- De acordo com a formulação de Wright, a evolução faz com que as populações se movam ao longo dos cenários por caminhos particulares e a adaptação pode ser vista como um movimento em direção a picos locais.
- Um pico local (ou ótimo local) não é necessariamente o ponto mais alto no cenário, mas qualquer pequeno movimento no sentido de nos afastarmos do pico local conduzirá a uma diminuição em fitness.
- Operações de cruzamento e mutação podem ser vistas como formas de mover uma população em um cenário definido pela função de fitness.

## Como GAs funcionam?

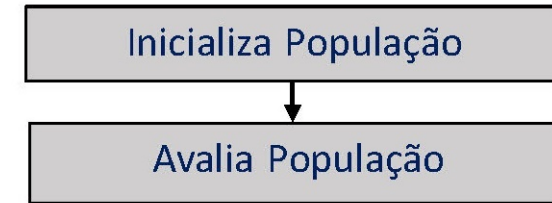
- GA cria população aleatória de indivíduos (soluções p/ o problema) - por serem aleatórias, nem sempre são boas.

Inicializa População

---

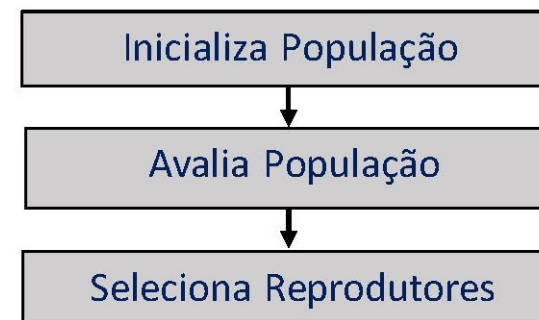
## Como GAs funcionam?

- GA cria população aleatória de indivíduos (soluções p/ o problema) - por serem aleatórias, nem sempre são boas.
- Submete a população a um processo que imita a evolução das espécies.



## Como GAs funcionam?

- GA cria população aleatória de indivíduos (soluções p/ o problema) - por serem aleatórias, nem sempre são boas.
- Submete a população a um processo que imita a evolução das espécies.



## Como GAs funcionam?

- GA cria população aleatória de indivíduos (soluções p/ o problema) - por serem aleatórias, nem sempre são boas.
- Submete a população a um processo que imita a evolução das espécies.



## Como GAs funcionam?

- GA cria população aleatória de indivíduos (soluções p/ o problema) - por serem aleatórias, nem sempre são boas.
- Submete a população a um processo que imita a evolução das espécies.





## Como GAs funcionam?

- GA cria população aleatória de indivíduos (soluções p/ o problema) - por serem aleatórias, nem sempre são boas.
- Submete a população a um processo que imita a evolução das espécies.



## Como GAs funcionam?

- GA cria população aleatória de indivíduos (soluções p/ o problema) - por serem aleatórias, nem sempre são boas.
- Submete a população a um processo que imita a evolução das espécies.
- A nova população gerada incluirá soluções melhores que as originais, que eram puramente aleatórias.
- As melhores soluções são adicionadas à população original, enquanto que as piores são eliminadas.



## Como GAs funcionam?

- GA cria população aleatória de indivíduos (soluções p/ o problema) - por serem aleatórias, nem sempre são boas.
- Submete a população a um processo que imita a evolução das espécies.
- A nova população gerada incluirá soluções melhores que as originais, que eram puramente aleatórias.
- As melhores soluções são adicionadas à população original, enquanto que as piores são eliminadas.
- O processo é repetido entre os melhores elementos, e melhores indivíduos serão novamente gerados, enquanto que os piores são eliminados...



## Como GAs funcionam?

- GA cria população aleatória de indivíduos (soluções p/ o problema) - por serem aleatórias, nem sempre são boas.
- Submete a população a um processo que imita a evolução das espécies.
- A nova população gerada incluirá soluções melhores que as originais, que eram puramente aleatórias.
- As melhores soluções são adicionadas à população original, enquanto que as piores são eliminadas.
- O processo é repetido entre os melhores elementos, e melhores indivíduos serão novamente gerados, enquanto que os piores são eliminados...



## Como GAs funcionam?

- GA cria população aleatória de indivíduos (soluções p/ o problema) - por serem aleatórias, nem sempre são boas.
- Submete a população a um processo que imita a evolução das espécies.
- A nova população gerada incluirá soluções melhores que as originais, que eram puramente aleatórias.
- As melhores soluções são adicionadas à população original, enquanto que as piores são eliminadas.
- O processo é repetido entre os melhores elementos, e melhores indivíduos serão novamente gerados, enquanto que os piores são eliminados...



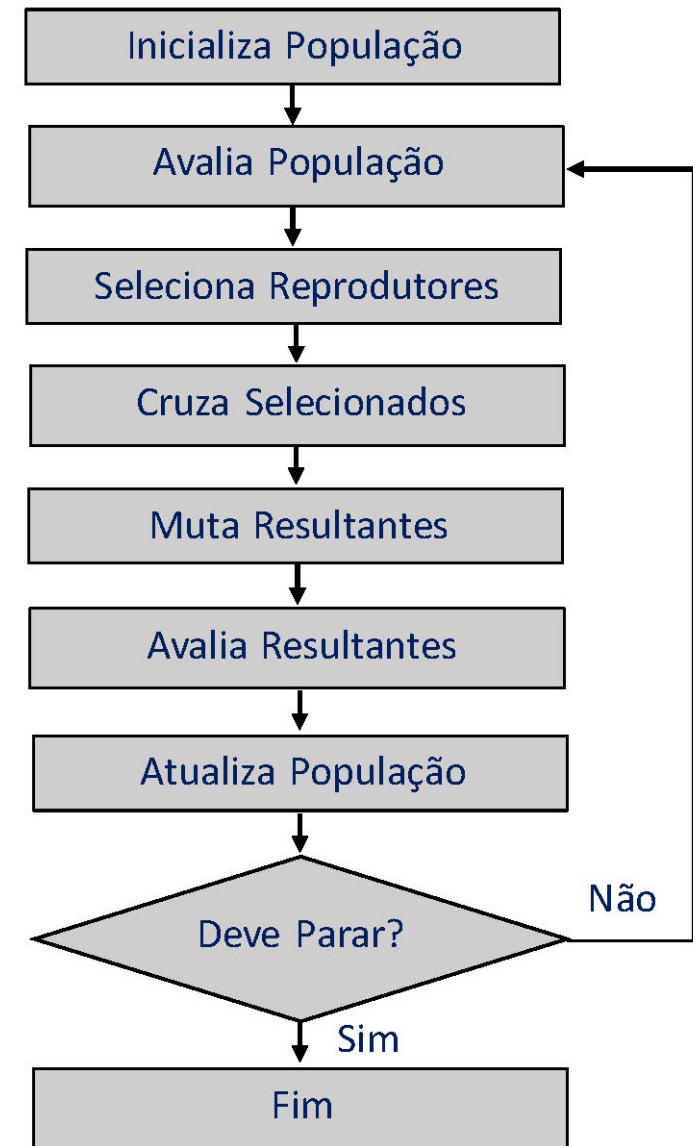
## Como GAs funcionam?

- GA cria população aleatória de indivíduos (soluções p/ o problema) - por serem aleatórias, nem sempre são boas.
- Submete a população a um processo que imita a evolução das espécies.
- A nova população gerada incluirá soluções melhores que as originais, que eram puramente aleatórias.
- As melhores soluções são adicionadas à população original, enquanto que as piores são eliminadas.
- O processo é repetido entre os melhores elementos, e melhores indivíduos serão novamente gerados, enquanto que os piores são eliminados...



## Como GAs funcionam?

- GA cria população aleatória de indivíduos (soluções p/ o problema) - por serem aleatórias, nem sempre são boas.
- Submete a população a um processo que imita a evolução das espécies.
- A nova população gerada incluirá soluções melhores que as originais, que eram puramente aleatórias.
- As melhores soluções são adicionadas à população original, enquanto que as piores são eliminadas.
- O processo é repetido entre os melhores elementos, e melhores indivíduos serão novamente gerados, enquanto que os piores são eliminados...

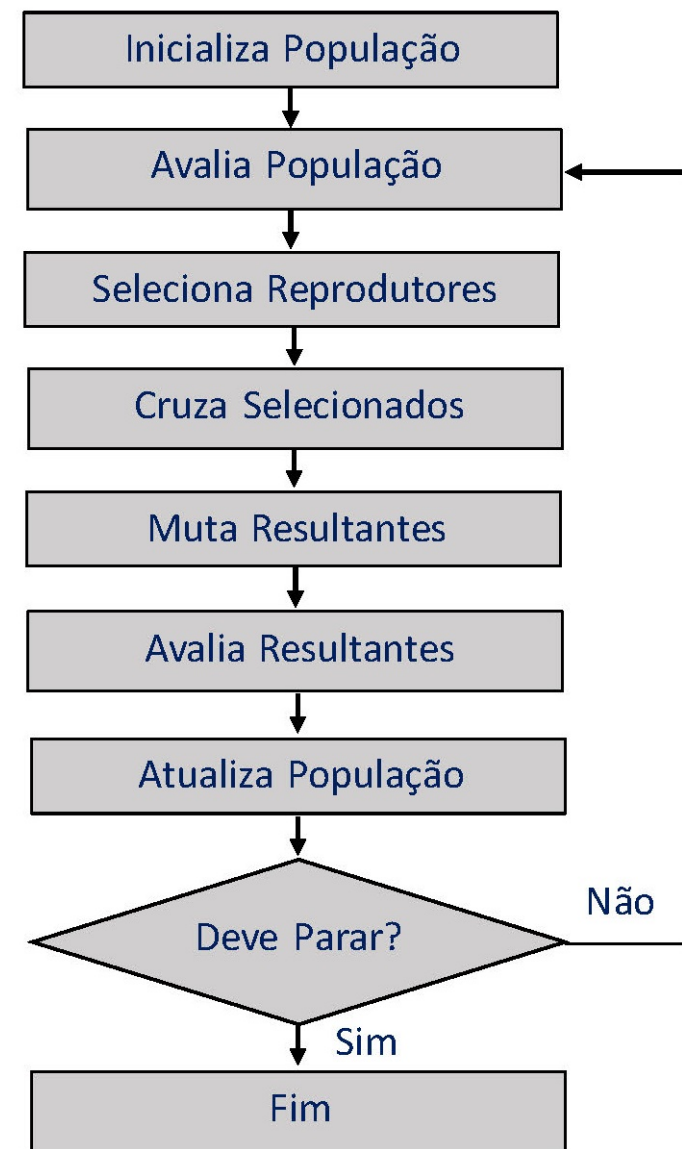


## Por que GAs funcionam ?

- O processo de cruzamento e seleção conduz a sempre melhores soluções.
- Como as piores soluções são descartadas, a população resultante será sempre melhor, em um sentido global.
- Este processo é chamado pressão seletiva.



*Algoritmos Genéticos:  
Seleção Natural Computacional*





## GA Binário x GA Contínuo

Binário	Contínuo																					
<ul style="list-style-type: none"> <li>Opera sobre palavras binárias que representam parâmetros (cromossomos) originalmente expressos em ponto flutuante.</li> </ul>	<ul style="list-style-type: none"> <li>Opera diretamente sobre parâmetros (cromossomos) expressos em ponto flutuante.</li> </ul>																					
<table border="1" style="margin-left: auto; margin-right: auto;"> <tbody> <tr><td>1010101010</td></tr> <tr><td>1110001000</td></tr> <tr><td>0011011100</td></tr> <tr><td>0100001111</td></tr> <tr><td>1101100011</td></tr> <tr><td>0011101010</td></tr> <tr><td>....</td></tr> </tbody> </table>	1010101010	1110001000	0011011100	0100001111	1101100011	0011101010	....	<table border="1" style="margin-left: auto; margin-right: auto;"> <thead> <tr> <th><math>x</math></th> <th><math>y</math></th> </tr> </thead> <tbody> <tr><td>9.0465</td><td>8.3097</td></tr> <tr><td>9.1382</td><td>5.2693</td></tr> <tr><td>7.6151</td><td>9.1032</td></tr> <tr><td>2.7708</td><td>8.4617</td></tr> <tr><td>8.9766</td><td>9.3469</td></tr> <tr><td>...</td><td>...</td></tr> </tbody> </table>	$x$	$y$	9.0465	8.3097	9.1382	5.2693	7.6151	9.1032	2.7708	8.4617	8.9766	9.3469	...	...
1010101010																						
1110001000																						
0011011100																						
0100001111																						
1101100011																						
0011101010																						
....																						
$x$	$y$																					
9.0465	8.3097																					
9.1382	5.2693																					
7.6151	9.1032																					
2.7708	8.4617																					
8.9766	9.3469																					
...	...																					

## GAs Binários – Elementos

- Os indivíduos em uma população são constituídos por um conjunto de palavras binárias, cada palavra binária constituindo um cromossomo (parâmetro).
- Cada palavra binária que constitui um cromossomo (parâmetro) resulta da conversão do parâmetro originalmente representado numericamente em ponto flutuante  $p/$  a sua representação inteira **normalizada**.
- Para minimizar erros de quantização, a **normalização** (= *Variable Encoding*) do parâmetro deve ser tal que o menor número inteiro corresponda ao menor número em ponto flutuante do universo de valores possíveis do parâmetro e o maior número inteiro corresponda ao maior número em ponto flutuante do universo de valores possíveis do parâmetro. Por exemplo, seja um cromossomo (parâmetro) cujo universo de valores possíveis varie no intervalo  $[-1.0, 1.0]$ , universo este a ser representado por palavras binárias de 16 bits. A normalização deve então obedecer ao mapeamento  $[-32768, 32767]_{(int)} = [1000000000000000, 0111111111111111]_{(bin)} \Rightarrow [-1.0, 1.0]_{(float)}$ . O bit em vermelho é o bit de sinal na representação inteira em complemento de 2.
- Cada indivíduo com seus cromossomos é um ponto localizado no espaço de busca de soluções candidatas. Bits no cromossomo constituem os genes.
- O GA opera iterativamente sobre a população de indivíduos, substituindo sucessivamente indivíduos de menor *fitness* por indivíduos de maior *fitness*.

População	
$n=4$ indivíduos com $s=3$ cromossomos (parâmetros) por indivíduo, cada cromossomo sendo de tamanho $\ell = 8$ bits	Fitness
00010101 00111010 11110000	0.67
00010001 00111011 10100101	0.23
00100100 10111001 01111000	0.45
11000101 01011000 01101010	0.94

## GAs Binários – Operadores

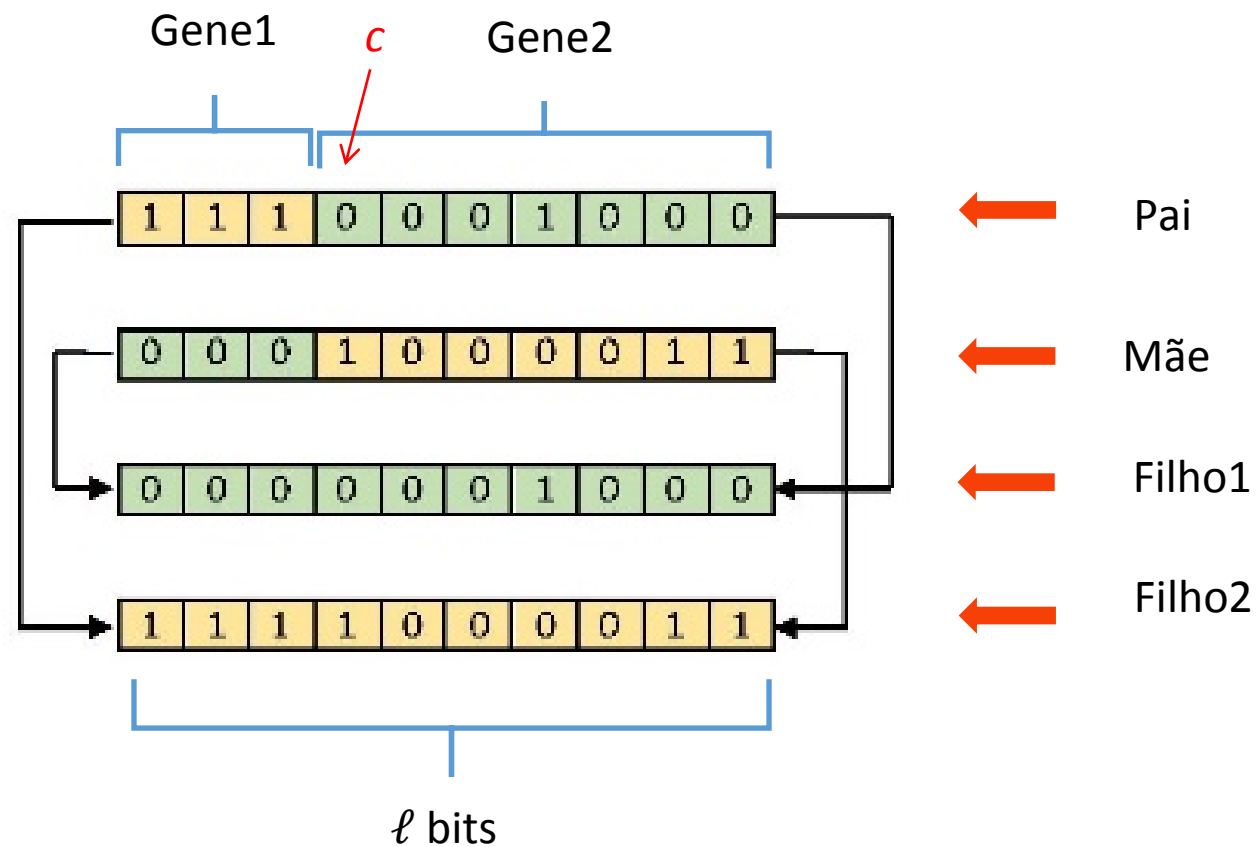
### Operador Seleção

- Seleciona a parcela  $p_c$  da população com os *maiores fitness*, sendo  $0.0 \leq p_c \leq 1.0$  ( $p_c$  é o *crossover rate*). Esta parcela  $p_c$  de indivíduos define o ***mating pool*** (tanque de pareamento) de indivíduos (pai e mãe) que gerarão por reprodução os indivíduos filhos (cada pai e mãe geram um par de filhos).
- **Quanto maior o *fitness* do indivíduo, mais vezes ele será selecionado para reprodução** (*Roulette Wheel Weighting – a ser visto a seguir*).
- Seleciona aleatoriamente um (ou mais) cromossomos (parâmetros) do indivíduo pai para o cruzamento (crossover) com o cromossomo homólogo (= em mesma posição) no indivíduo mãe.

População		Fitness
1   0   1   1   0   0   1   1   0   1		3.667899
1   1   1   0   0   0   1   0   0   0	← Pai	9.261380
0   1   1   0   0   1   0   1   0   1		3.481746
0   0   0   1   0   0   0   0   1   1	← Mãe	12.864222

## Operador Cruzamento (*Crossover*)

- Seleciona aleatoriamente uma posição  $c$  e troca os bits (genes) nas posições anteriores e posteriores à  $c$  nos cromossomos pai e mãe. Note que a operação ocorre entre 2 cromossomos homólogos selecionados dentre os  $s$  cromossomos respectivos ao pai e mãe que foram selecionados previamente pelo *Roulette Wheel*. O resultado é a criação de 2 indivíduos filhos:



## Operador Mutação (*Mutation*)

- Inverte aleatoriamente alguns poucos bits nos cromossomos de indivíduos.
- Mutação pode ocorrer em um (ou mais) bit de uma palavra binária, com taxa de mutação (*mutation rate*) usualmente muito pequena (da ordem de 3% a 15% do número de indivíduos).

0	0	0	0	0	1	0	0
---	---	---	---	---	---	---	---

0	0	0	0	0	1	0	0
---	---	---	---	---	---	---	---



0	1	0	0	0	1	0	0
---	---	---	---	---	---	---	---

## Algoritmo Genético Binário Básico

Sendo dado um problema claramente definido para ser resolvido e palavras binárias de tamanho  $\ell$  representando soluções candidatas, um **algoritmo genético básico** irá operar conforme segue:

1. Iniciar com uma população gerada aleatoriamente de  $n$  ( $n$  par) indivíduos com  $s$  cromossomos de  $\ell$  bits (soluções candidatas).
2. Calcular o fitness  $f(x)$  de cada indivíduo  $x$  presente na população e reordenar os indivíduos (polinômios) em ordem crescente de *fitness*. Se algum dos *fitness* for suficientemente alto, termina o processo.
3. Repetir os seguintes passos até que  $n$  indivíduos tenham sido criados:
  - 3.1 Selecionar a parcela  $p_c$  da população com os *maiores fitness*, sendo  $0.0 \leq p_c \leq 1.0$  ( $p_c$  é o *crossover rate*). Esta parcela de indivíduos define o *mating pool* (tanque de pareamento).
  - 3.2 Selecionar um par de indivíduos do *mating pool* para desempenhar a função de pai e mãe. A probabilidade de seleção deve ser uma função crescente do *fitness* (por exemplo, o *Roulette Wheel Weighting* – a ser visto a seguir). No processo de seleção o mesmo indivíduo e o mesmo cromossomo podem ser selecionados para ser pai (ou mãe) mais do que uma vez.

3.2 Executar o cruzamento (*crossover*) a partir de uma posição  $c$  aleatória ( $1 \leq c \leq \ell$  c/ probabilidade uniforme) em um dos  $\ell$  bits dos cromossomos pai e filho, e com isto formar 2 indivíduos filhos através da troca dos bits (genes) nas posições anteriores e posteriores àquela selecionada nos cromossomos pai e mãe (conforme visto em slide anterior). Pode-se executar o *crossover* em todos os  $s$  pares de cromossomos homólogos, ou apenas em um (ou mais) par de cromossomos homólogos selecionados aleatoriamente. Após o *crossover* os indivíduos pai e mãe são descartados sendo substituídos pelos dois filhos.

3.3 Selecionar a parcela  $p_m$  dos indivíduos resultantes no *mating pool* após o *crossover*, sendo  $0.0 < p_m < 1.0$  ( $p_m$  é o *mutation rate*). Inverter um dos  $\ell$  bits de um dos  $s$  cromossomos (aleatoriamente e com probabilidade uniforme) de cada indivíduo selecionado.

4. Adicionar à população do *mating pool* os indivíduos da população anterior que não foram selecionados para o *mating pool*.

5. Voltar ao passo 2.



## Seleção de indivíduos pelo *Roulette Wheel Weighting*

**Método utilizado:** Probabilidade de seleção proporcional ao *fitness* do indivíduo.

Este critério é equivalente à “seleção por viabilidade”, em biologia. Neste tipo de seleção, o número de vezes que um indivíduo poderá se reproduzir é proporcional à razão:

$$\frac{\textit{fitness do indivíduo } k}{\text{somatório dos } \textit{fitness de todos os indivíduos da população}}$$

O critério de seleção pelo *fitness* pode ser implementado conforme Figura 2. Consiste em atribuir a cada indivíduo uma “fatia” (de uma roleta) que corresponda em área ao *fitness* do indivíduo.

A roleta é girada e a “bolinha” repousa sobre uma das “fatias”. Como o tamanho da população é  $n = 4$ , a roleta será girada 4 vezes, podendo escolher, por exemplo:

- |                         |   |                           |
|-------------------------|---|---------------------------|
| 1º. Giro → Cromossomo B | } | Indivíduos pais sorteados |
| 2º. Giro → Cromossomo D |   |                           |
| 3º. Giro → Cromossomo B | } | Indivíduos pais sorteados |
| 4º. Giro → Cromossomo C |   |                           |

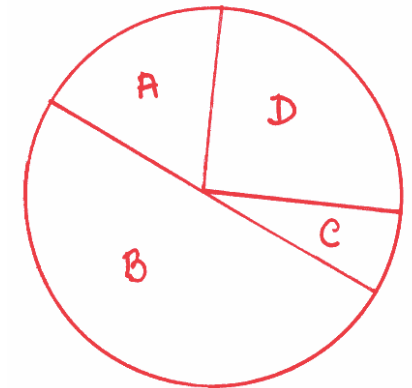
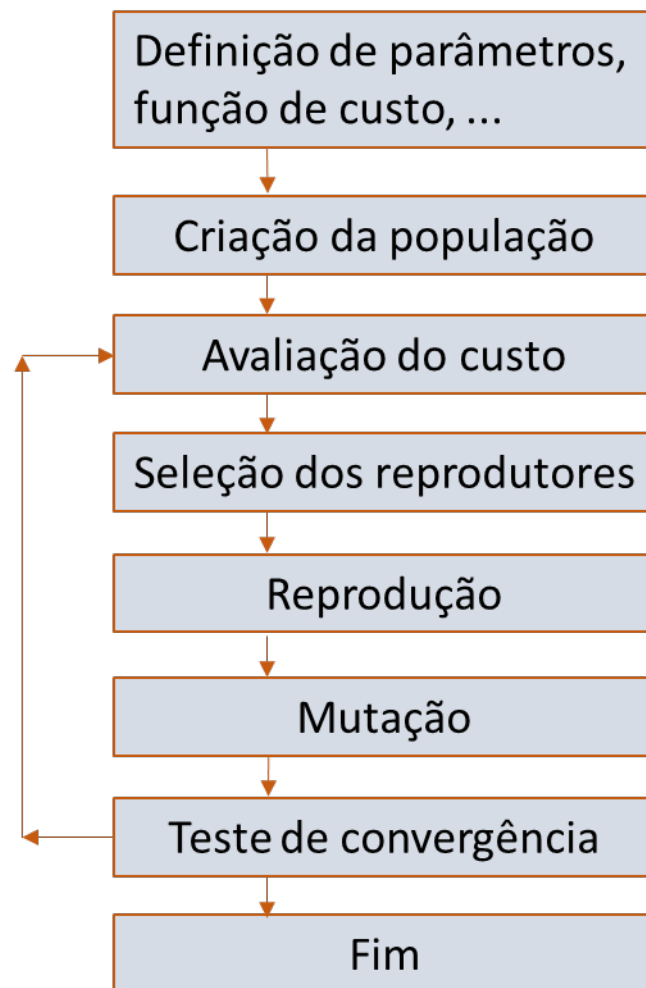


Figura 2: “Roulette-Wheel Sampling” (Goldberg, 1989)

Note que se a roleta for girada muitas vezes, os resultados médios estarão próximos aos esperados.

## GAs de Parâmetros Contínuos



\* Os parâmetros não são representados por zeros e uns (não há o bloco “representação paramétrica”)

\* Os parâmetros são representados por números reais, sobre qualquer intervalo que seja apropriado.

- Os vetores de parâmetros (indivíduos) são vetores  $N_{par}$  - *dimensionais*, da forma:

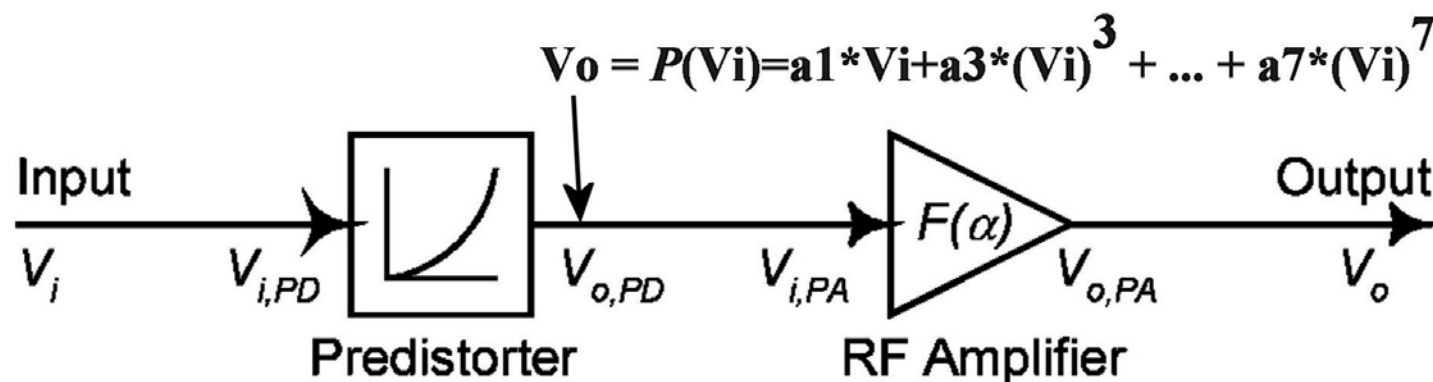
$$\text{indivíduo} = [p_1, p_2, \dots, p_{N_{par}}]$$

- Os **parâmetros (cromossomos)**  $p_1, p_2, \dots, p_{N_{par}}$  são representados por **números em ponto flutuante (números reais)**.
- Cada indivíduo da população tem o a sua aptidão avaliada da função de *fitness*  $f$  cujos argumentos são os parâmetros  $p_1, p_2, \dots, p_{N_{par}}$ :

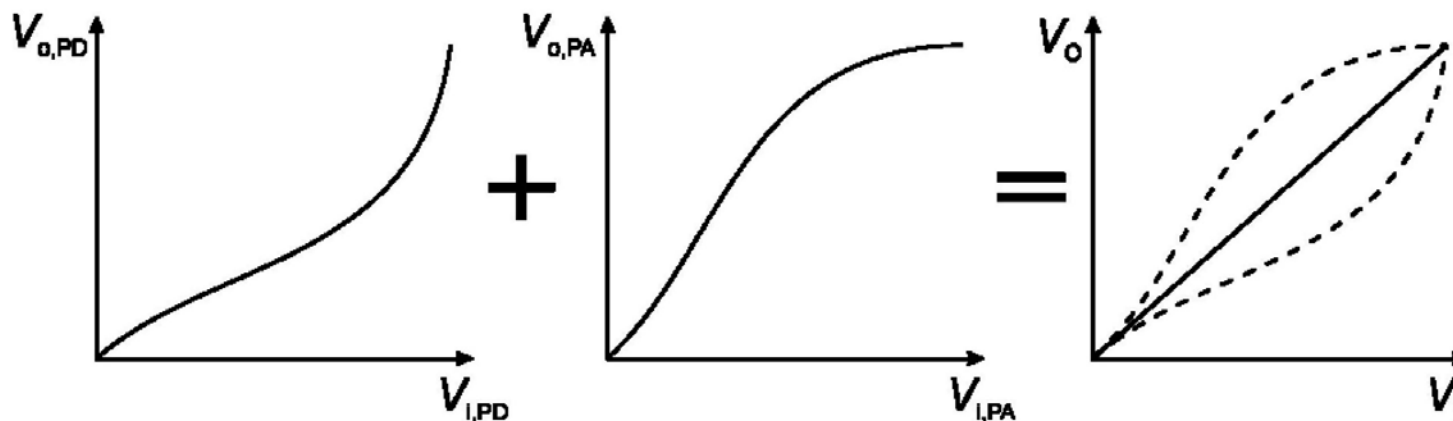
$$\text{fitness} = f(\text{indivíduo}) = f(p_1, p_2, \dots, p_{N_{par}})$$

- Tendo sido gerados os indivíduos, os parâmetros (cromossomos) individuais são passados para a função de *fitness*, para a determinação da aptidão de cada indivíduo da população.

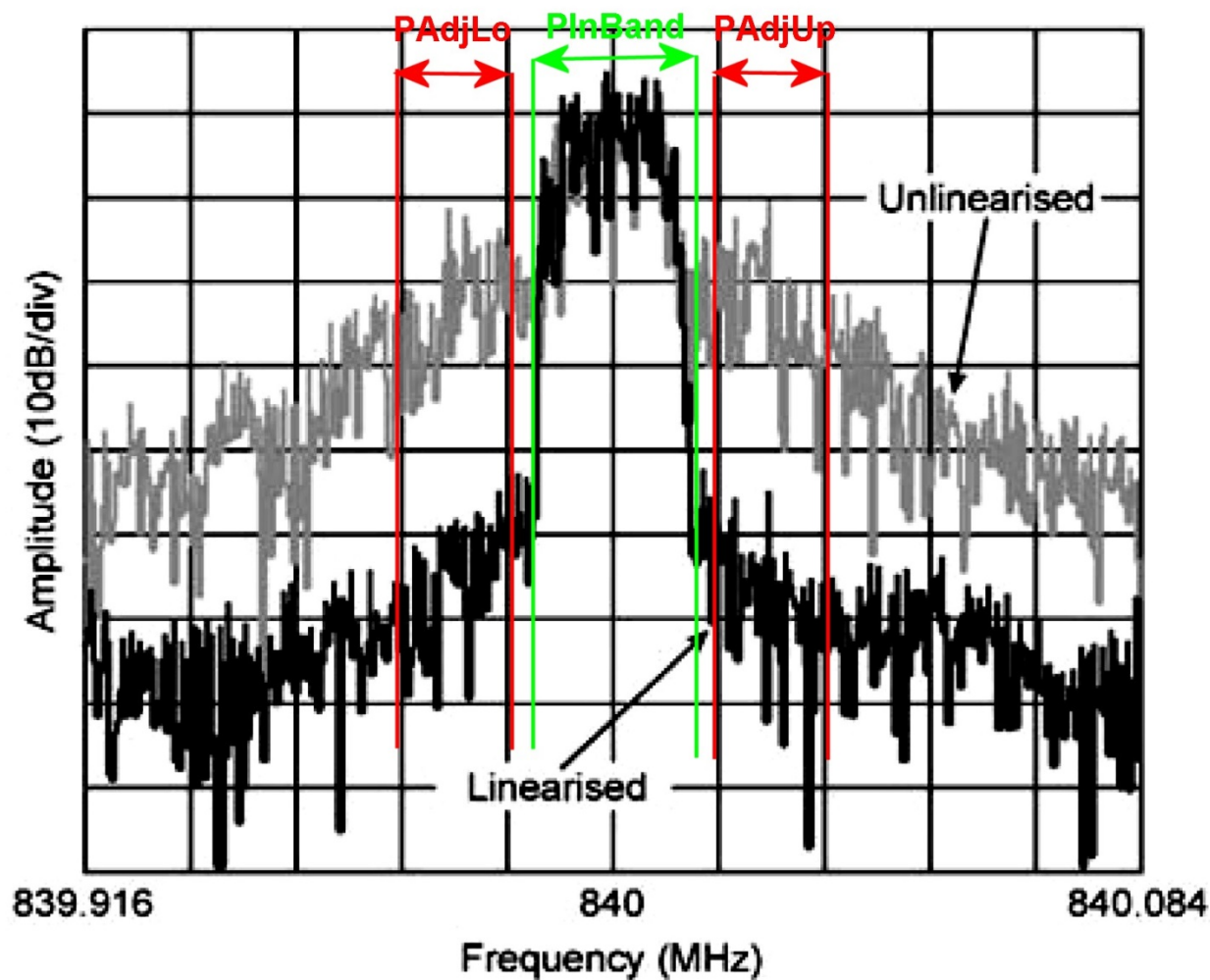
## Exemplo - Linearização de HPA de RF através de GA contínuo



Objetivo: Determinar os coeficientes  $a_1, a_3, \dots, a_7$  do polinômio  $P(V_i)$  de forma a que  $P(V_i)$  seja a função inversa do polinômio  $\alpha(V_i)$  que descreve a não-linearidade do HPA, linearizando, portanto, a transmitância conjunta do Predistorter + HPA:

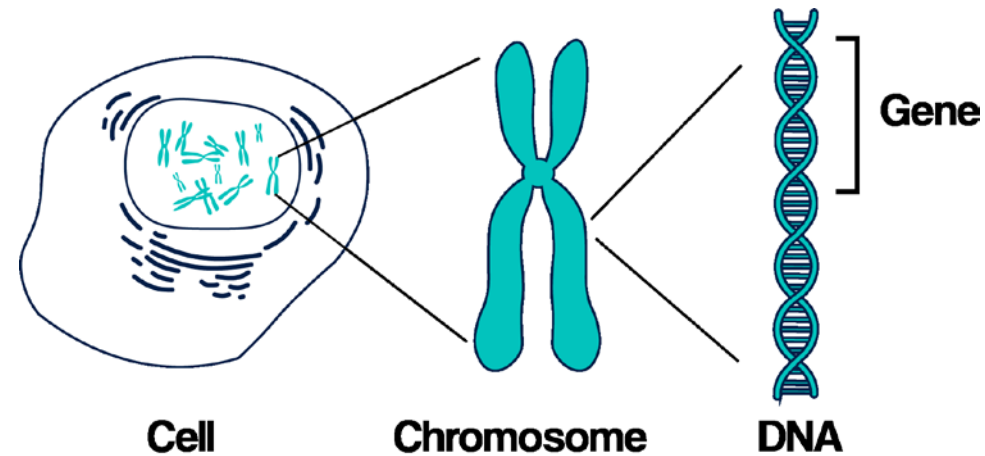


Indicador do grau de linearização (*fitness* para o GA):  $Fitness = 1 - ACPR$ , onde o  $ACPR$  (*Adjacent Channel Power Ratio*) é obtido do espectro de saída do HPA conforme abaixo:



$$ACPR = \frac{PAdjLo + PAdjUp}{PInBand}$$

## Indivíduo, cromossomos e genes constituintes:



- Cada indivíduo da população é um polinômio de 4 coeficientes ( $a_1, a_3, \dots, a_7$ ).
- Cada coeficiente (=cromossomo) é representado por um número em ponto flutuante.
- Não há estrutura formal para os genes - os mesmos ficam virtualmente (i.e., implicitamente) definidos na operação de *crossover* (a ser vista a seguir).

Indivíduos	Operação
<p>[0.7530, 1.01250, -0.025, -0.008]            [0.9000, 0.18758, -0.010, -0.001]            [0.8100, 2.69740, -0.015, -0.002]            ...</p>	<ul style="list-style-type: none"> <li>• População de indivíduos (polinômios) é inicializada com valores aleatórios em cada cromossomo (coeficiente do polinômio), <b>mas de valores pertinentes ao problema em questão.</b></li> <li>• Cada vetor em ponto flutuante ao lado representa os 4 cromossomos (coeficientes) de um indivíduo (polinômio) na população.</li> <li>• <i>Fitness</i>: Função que caracteriza a linearização do HPA (<math>\text{Fitness}=1 - \text{ACPR}</math>) para cada indivíduo (polinômio).</li> <li>• Desejamos reproduzir os indivíduos (polinômio) com maior <i>fitness</i>.</li> </ul>
<p>[0.9000, 0.18758, -0.010, -0.001]            [0.8100, 2.69740, -0.015, -0.002]            [0.7530, 1.01250, -0.025, -0.008]            ...</p>	<ul style="list-style-type: none"> <li>• Avaliação do <i>fitness</i> da população (seleção de potenciais reprodutores)               <ul style="list-style-type: none"> <li>– Avalia-se o <i>fitness</i> de cada indivíduo (polinômio) da população inteira e reordena-se os indivíduos (polinômios) em ordem crescente de <i>fitness</i>.</li> <li>– Apenas indivíduos (polinômios) com <i>fitness</i> maior que um determinado <i>threshold</i> determinado experimentalmente são selecionados para cruzamento (por exemplo, os ~20% da população de indivíduos com maior <i>fitness</i> <math>\Rightarrow</math> <i>crossover rate</i> = 0.2), definindo o <b><i>mating pool</i></b>.</li> </ul> </li> </ul>
<p>[0.9000, 0.18758, -0.010, -0.001]            [0.8100, 2.69740, -0.015, -0.002]            [0.7530, 1.01250, -0.025, -0.008]            ...</p>	<ul style="list-style-type: none"> <li>• Cruzamento (<i>crossover</i> - reprodução)               <ul style="list-style-type: none"> <li>– Pares de indivíduos do <i>mating pool</i> são aleatoriamente selecionados através do <b>RWW - Roulette Wheel Weighting</b> (slide 33). Cada par de indivíduos (polinômios) selecionados pelo RWW gera dois novos filhos (polinômios) através da troca de material genético entre cada correspondente cromossomo (=coeficiente=parâmetro). Um par de cromossomos (coeficientes=parâmetros) respectivos a um par de indivíduos (polinômios) pais é aleatoriamente selecionado e ocorre a troca de genes “virtuais” entre estes cromossomos conforme detalhado na seção “<i>Crossover</i>” a seguir.</li> </ul> </li> </ul>

## Crossover (cruzamento) em GAs contínuos

Da mesma forma que nos GAs binários, 2 indivíduos (polinômios) pais são escolhidos pelo **Roulette Wheel Weighting (RWW)** e dois filhos são gerados pela troca de genes “virtuais” entre os pais. Cada indivíduo (polinômio) é constituído por  $N_{par}$  cromossomos (=coeficientes= parâmetros). No caso exemplo da linearização do HPA,  $N_{par} = 4$ .

- Dentre as muitas heurísticas para cruzamento em GAs de parâmetros contínuos a mais frequentemente utilizada busca mimetizar a técnica adotada em GAs binários, da forma mais próxima possível:

Um índice aleatório $1 < \alpha < N_{par}$ é gerado definindo a posição $\alpha$ do par de cromossomos (=coeficientes= parâmetros) nos dois indivíduos pais $p_1$ e $p_2$ selecionados pelo RWW .para	$\alpha = \text{roundup} \{ \text{random} * N_{par} \}$
Sejam os individuos pais $p_1$ e $p_2$ , em que os sub-escritos $m$ e $p$ referem-se, respectivamente, à mãe e pai, conforme explicitado ao lado.	$p_1 = [p_{m1} \ p_{m2} \ \dots \ p_{m\alpha} \ \dots \ p_{mN_{par}}]$ $p_2 = [p_{p1} \ p_{p2} \ \dots \ p_{p\alpha} \ \dots \ p_{pN_{par}}]$
O par de cromossomos (=coeficientes= parâmetros) selecionados na posição $\alpha$ são combinados para formar novos cromossomos na posição $\alpha$ dos indivíduos cromossomos filhos conforme o parâmetro de miscigenação $\beta$ , que é um valor aleatório entre 0 e 1:	$p_{novo1} = p_{m\alpha} - \beta [p_{m\alpha} - p_{p\alpha}]$ $p_{novo2} = p_{p\alpha} + \beta [p_{m\alpha} - p_{p\alpha}]$
O <i>crossover</i> é então completado preenchendo os vetores com os demais cromossomos (=coeficientes= parâmetros) originais que não foram selecionados pelo índice $\alpha$ :	filho 1 = $[p_{m1} \ p_{m2} \ \dots \ p_{novo1} \ \dots \ p_{pN_{par}}]$ filho 2 = $[p_{p1} \ p_{p2} \ \dots \ p_{novo2} \ \dots \ p_{mN_{par}}]$



Indivíduos	Operação
<p>[0.9000, 0.18758, -0.010, -0.001]  [0.8100, 2.69740, -0.015, -0.002]</p>	<ul style="list-style-type: none"> <li>• Cruzamento (<i>crossover</i> - reprodução) - <b>CONTINUAÇÃO</b> <ul style="list-style-type: none"> <li>– Os indivíduos selecionados como pais p/ o cruzamento são aleatoriamente selecionados em pares através do <b>RWW - Roulette Wheel Weighting</b> (slide 49) . Cada par de indivíduos (polinômios) selecionados pelo RWW gera dois novos filhos (polinômios) através da troca de material genético entre cada correspondente cromossomo (=coeficiente=parâmetro). Um par de cromossomos (coeficientes=parâmetros) respectivos a um par de indivíduos (polinômios) pais é aleatoriamente selecionado e ocorre a troca de genes “virtuais” entre estes cromossomos conforme visto na seção “Crossover”.</li> <li>– Por exemplo, seja <math>\alpha=2</math> e <math>\beta= 0.272</math> para os indivíduos pais mostrados ao lado. Os dois filhos são: <div style="margin-left: 40px;"> <p>Filho1 = [0.9000, <math>p_{novo1}</math>, -0.010, -0.001]</p> <p>Filho2 = [0.8100, <math>p_{novo2}</math>, -0.015, -0.002]</p> </div> </li> </ul> </li> </ul> <p>onde</p> $p_{novo1} = p_{ma} - \beta [p_{ma} - p_{pa}] = 2.6974 - 0.272 * (2.6974 - 0.18758) = 2.015$ $p_{novo2} = p_{pa} + \beta [p_{ma} - p_{pa}] = 0.18758 + 0.272 * (2.6974 - 0.18758) = 0.87$
<p>[0.9000, 2.015, -0.010, -0.001]  [0.8100, 0.87, -0.015, -0.002]  ...</p>	<ul style="list-style-type: none"> <li>• Prole <ul style="list-style-type: none"> <li>– Os polinômios-filho têm alta probabilidade de linearizarem o HPA, porque ambos os polinômios-pais têm cromossomos que resultaram em um alto <i>fitness</i>.</li> <li>– Os novos parâmetros (cromossomos) dos filhos resultam de uma combinação linear dos cromossomo de ambos os pais.</li> </ul> </li> </ul>
<p>[0.9000, 2.015, -0.010, -0.001]  [0.8100, 0.87, -0.015, -0.012]  ....</p>	<ul style="list-style-type: none"> <li>• Mutação <ul style="list-style-type: none"> <li>– Para evitar o “elitismo” na seleção, seleciona-se aleatoriamente uma pequena parcela (~3% <math>\Rightarrow</math> mutation rate = 0.03) de indivíduos (polinômios) da população.</li> <li>– Para estes indivíduos selecionados um índice aleatório <math>1 &lt; \alpha &lt; N_{par}</math> é gerado definindo a posição <math>\alpha</math> do cromossomo (=coeficiente= parâmetro) que sofrerá mutação.</li> </ul> </li> <li>• A mutação consiste em substituir o valor do parâmetro (=coeficiente=cromossomo) por um valor gerado aleatoriamente <b>dentro da faixa de valores pertinentes ao problema em questão.</b></li> </ul>

Indivíduos	Operação
[0.9000, 2.015, -0.010, -0.001] [0.8100, 0.87, -0.015, -0.012] ....	<ul style="list-style-type: none"><li>• Nova População<ul style="list-style-type: none"><li>– Cada dois novos filhos substituirão os dois pais, portanto não há alteração no número de indivíduos da população</li><li>– Volta-se à etapa “Avaliação do fitness da população” e o processo é executado recorrentemente até que o Fitness=1 – ACPR de um indivíduo (polinômio) da população seja próximo de 1.0 de modo que o ACPR é próximo de zero, indicando que o polinômio lineariza o HPA.</li></ul></li></ul>

## Vantagens de GAs sobre outros algoritmos de otimização

- Otimizam com parâmetros discretos ou contínuos;
- Não requerem cálculos de derivada;
- Buscam simultaneamente pelo mínimo, a partir de uma amostragem ampla da superfície de custo;
- Lidam com grande número de parâmetros;
- São adequados para computação paralela (avaliar o *fitness* em paralelo, por exemplo);
- Capaz de otimizar parâmetros com funções de custo altamente complexas;
- Capaz de pular fora de um mínimo local (mutação);
- Conseguem prover uma lista de parâmetros ótimos, não apenas uma única solução;
- Podem codificar os parâmetros, de tal forma que a otimização seja feita com os parâmetros codificados;
- Trabalham com dados numericamente gerados, dados experimentais ou funções analíticas.