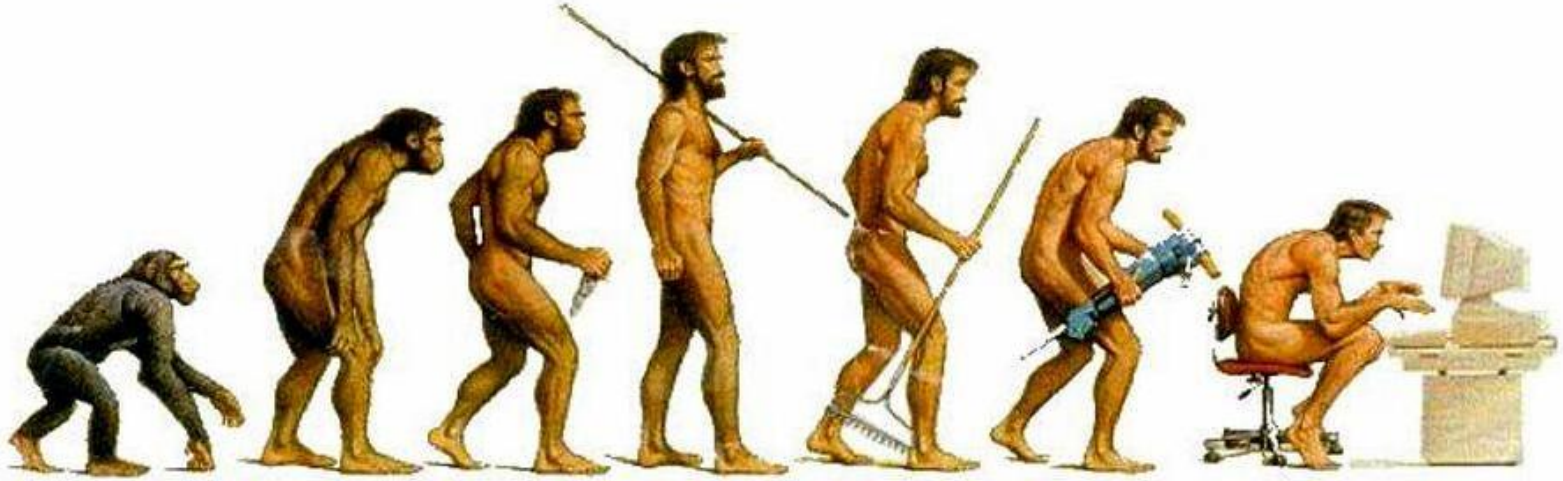
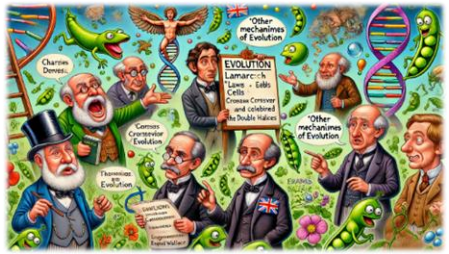
Computação Evolutiva

Parte 2



Fabricio Breve – www.fabriciobreve.com

Aula Anterior: Biologia Evolutiva

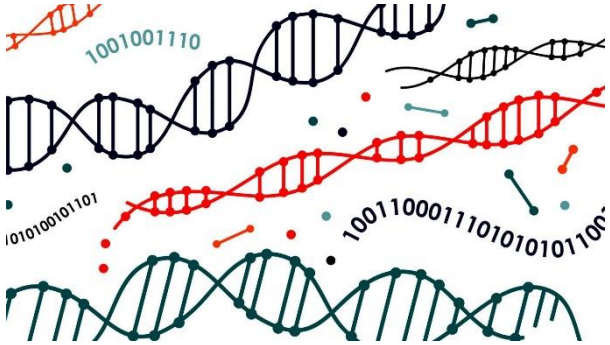


- História
- Lamarck
- Leis de Lamarck
- Princípios do Lamarckismo
- Charles Darwin
- A Viagem de Darwin
- Erasmus Darwin
- Alfred Wallace
- Teoria da Evolução
- Mendel
- Experimentos de Mendel
- Leis de Mendel
- Genética
- Células
- DNA
- Cross-Over
- Mutação
- Evolução como Resultado de Variação Genética mais Seleção
- Outros Mecanismos de Evolução
- O Caso da Obesidade em Samoa
- Exemplo Clássico das Mariposas na Inglaterra
- Questões

Aula de Hoje

- Principais Características
- Algoritmo Evolutivo Padrão
- Algoritmos Genéticos
- Indivíduo
- Exemplo: Preparo de Biscoito
- Codificação
  - Representações
  - Corte e Escala
  - Código de Gray
  - Código Termômetro
- Função de Aptidão
  - Função de Aptidão Padrão
  - Aptidão Baseada em Ranking
  - Aptidão Ranking-Espaço
- Seleção
  - Seleção pela Roleta
  - Seleção por Torneio
  - Pressão Seletiva
  - Diversidade
  - Seleção por Amostragem Universal Estocástica
  - Seleção por Estado Estável
- Operadores Genéticos
  - Crossover
    - Crossover de Um Ponto
    - Crossover de Dois Pontos
    - Crossover Uniforme
  - Mutação
  - Elitismo
- Funcionamento do Algoritmo Genético
- Critério de Parada
  - Convergência
- Exemplos:
  - Reconhecimento de Padrões
  - Minimização de Polinômio de 4º Grau
- Aplicações
- Exercício
- Comparação: Subida da Colina, Recozimento Simulado e Algoritmos Genéticos
  - Analogia com Cangurus
- Bibliografia

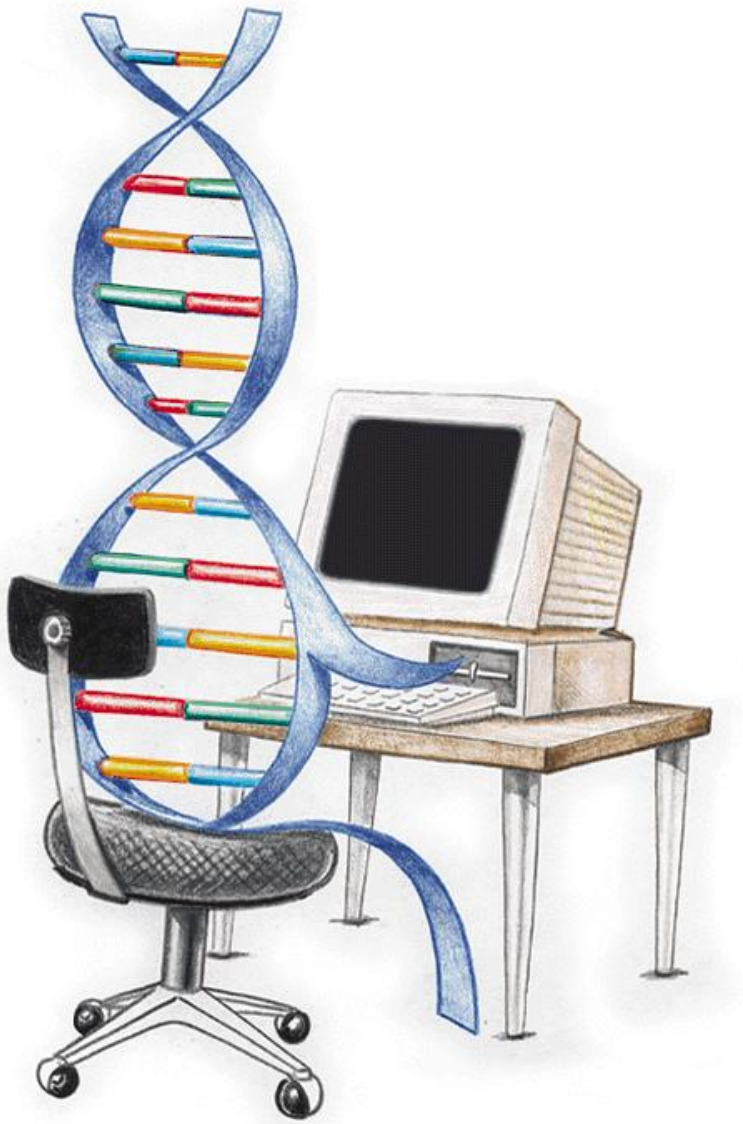
Computação Evolutiva



- Evolução pode ser vista como:
  - Processo capaz de localizar soluções para problemas oferecidos pelo ambiente.
- *Algoritmos Evolutivos*
  - Algoritmos inspirados na evolução.
- *Computação Evolutiva*
  - Campo de pesquisa que engloba todos os algoritmos evolutivos.

Computação Evolutiva

- Principais categorias:
  - Algoritmos Genéticos
  - Estratégias Evolutivas
  - Programação Evolutiva
  - Programação Genética



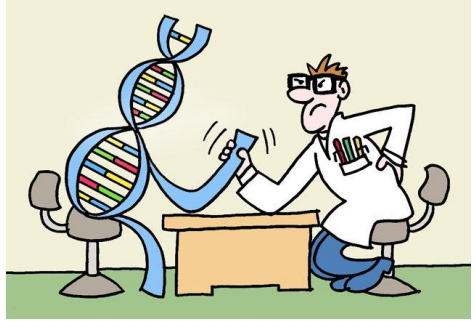
Computação Evolutiva

- Estratégias Evolutivas (1965)
  - Inicialmente usadas para resolver problemas de mecânica dos fluidos e posteriormente generalizadas para resolver problemas de otimização de funções, focando o caso de funções de valor real.
  - Os primeiros modelos operavam com um único indivíduo da população submetido apenas à mutação.
- Programação Evolutiva (1962)
  - Proposta com o objetivo de utilizar os conceitos de evolução no desenvolvimento da Inteligência Artificial.
  - Cada indivíduo da população é representado por uma máquina de estados finitos (MEF), que processa uma sequência de símbolos.

Computação Evolutiva

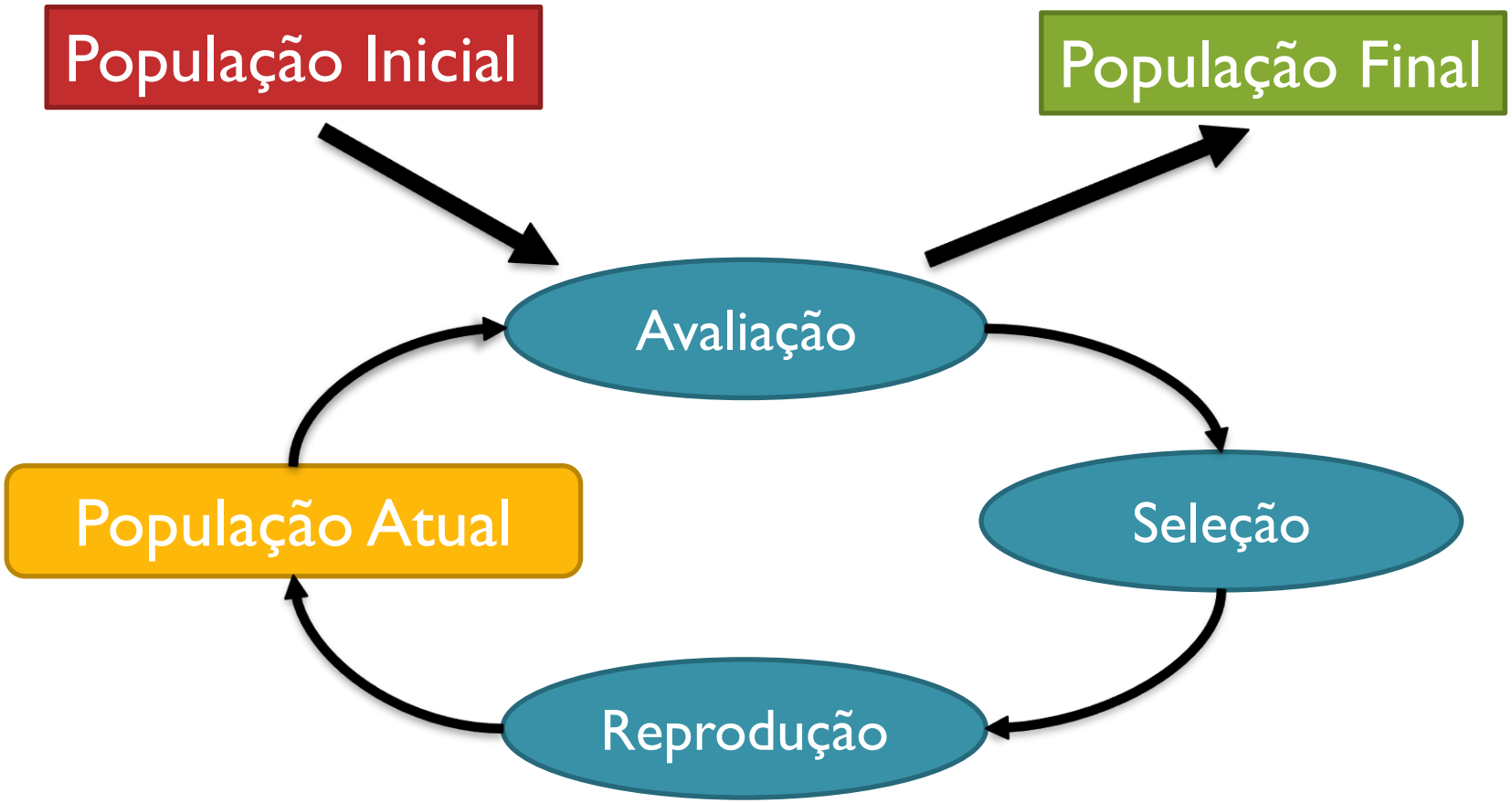
- Programação Genética (1985)
  - Constitui um tipo de algoritmo evolutivo desenvolvido como uma extensão dos algoritmos genéticos.
  - As estruturas de dados que sofrem adaptação são representações de programas de computador, e assim a avaliação de aptidão envolve a execução dos programas.
  - Envolve uma busca baseada na evolução do espaço de possíveis programas de computador de forma que, quando executados, produzam uma saída adequada, que geralmente está relacionada à resolução de um determinado problema.

Algoritmo Evolutivo Padrão

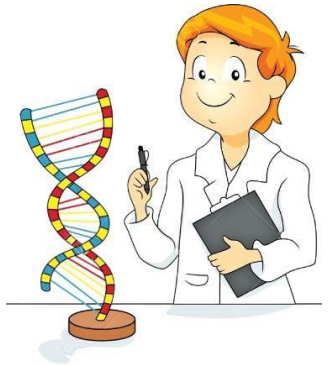


- É composto de:
  - Uma população de indivíduos que se reproduz com herança
    - Cada indivíduo codifica um ponto no espaço de busca e solução potencial para o problema.
    - Podem se reproduzir, gerando filhos que herdam algumas características dos pais.
  - Variação Genética
    - Filhos sujeitos a variação genética por mutação.
      - Permite o surgimento de novas características e a exploração de novas regiões do espaço de busca.
  - Seleção Natural
    - Avaliação dos indivíduos em seu ambiente resulta em uma medida de aptidão / adaptabilidade / qualidade.
      - O valor de aptidão de cada indivíduo é usado na competição por sobrevivência e reprodução.
        - Indivíduos com maior aptidão tem mais chances de serem selecionados.

Algoritmo Evolutivo Padrão



Algoritmo Evolutivo Padrão



- Genérico, Iterativo e Probabilístico.
- Mantém população **P** de *N* indivíduos $\mathbf{P} = \{\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_N\}$ em cada iteração *t*
- Cada indivíduo é uma solução potencial para o problema a ser resolvido.
- Indivíduos são avaliados em sua medida de adaptação ao ambiente, ou *aptidão*.
- Então, uma nova população é gerada na iteração $t + 1$ *selecionando* alguns (geralmente os mais aptos) dos indivíduos da população e *reproduzindo-os* sexuadamente ou assexuadamente.

Algoritmo Evolutivo Padrão

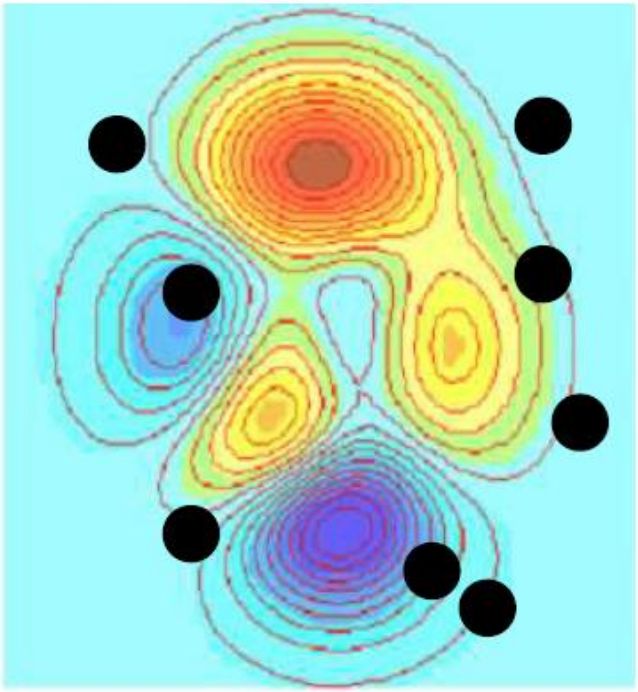
- Se for usada a reprodução sexuada, um operador de *recombinação genética (crossover)* pode ser usado.
- Variações genéticas através de *mutação* também podem afetar alguns indivíduos.
- Este processo todo é iterado.
- O conjunto destes passos (reprodução, variação genética, e seleção) constitui o que é chamado uma *geração*.

Algoritmo Evolutivo Padrão

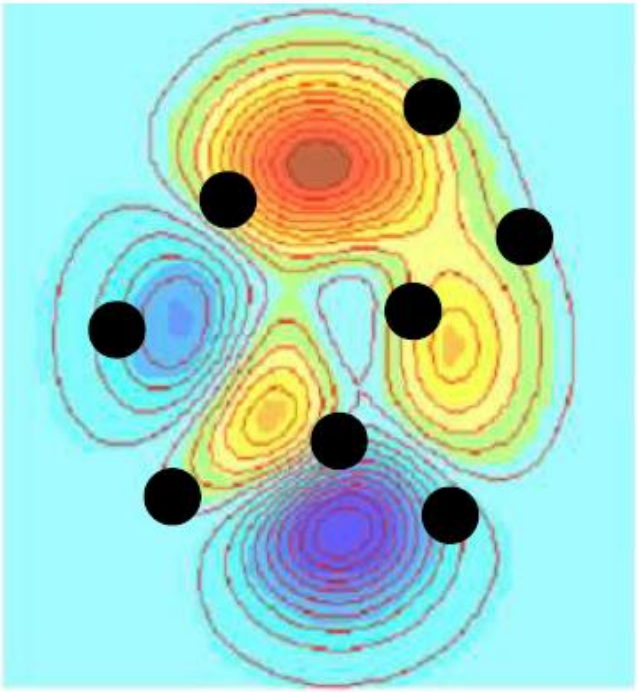
```
procedimento [P] = AE_padrao(pc,pm)
  inicializar P
  f ← avaliar(P)
  t ← 1
  enquanto não atender critério de parada faça
    P ← selecionar(P,f)
    P ← reproduzir(P,f,pc)
    P ← variar(P,pm)
    f ← avaliar(P)
    t ← t + 1
  fim-enquanto
fim-procedimento
```

pc – probabilidade de *crossover*
pm – probabilidade de mutação

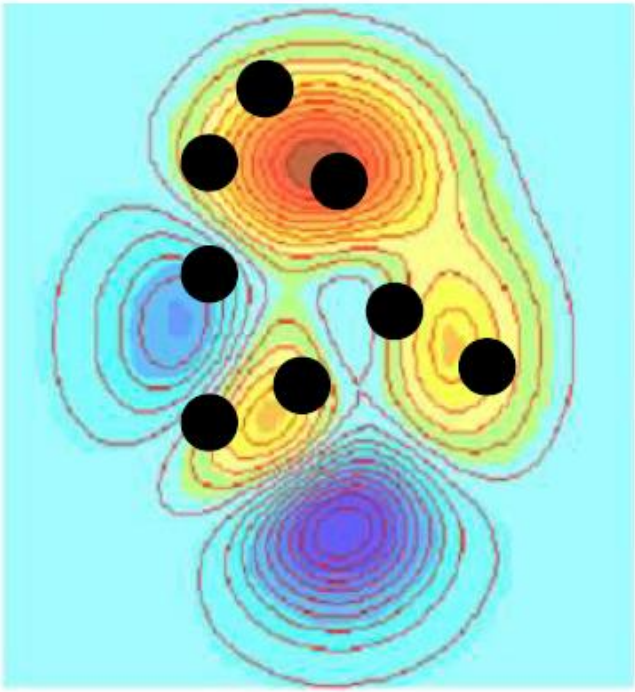
Processo Evolutivo Padrão



Geração 1

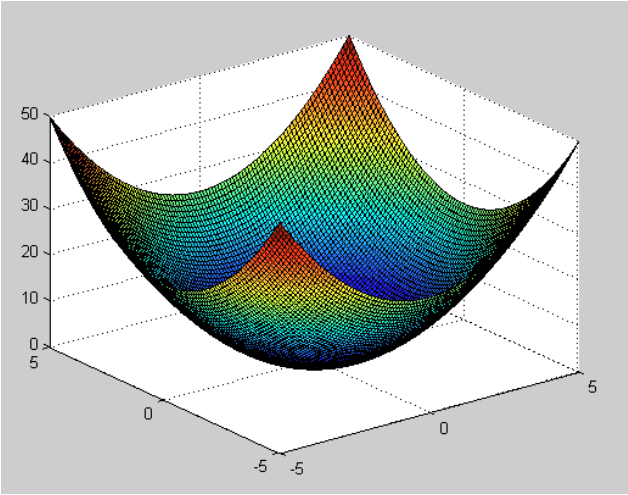


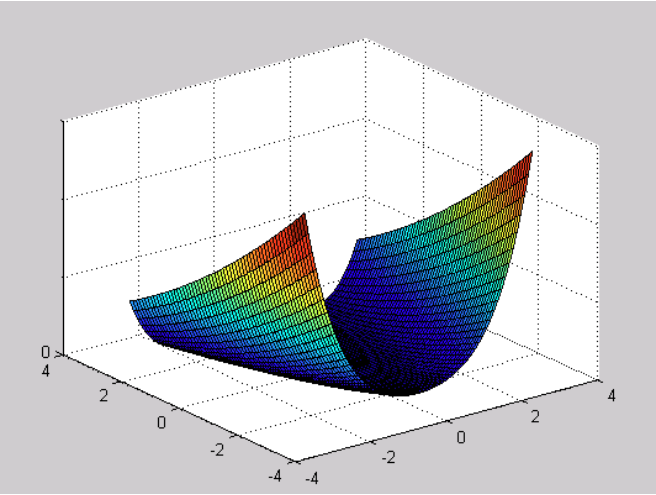
Geração 10

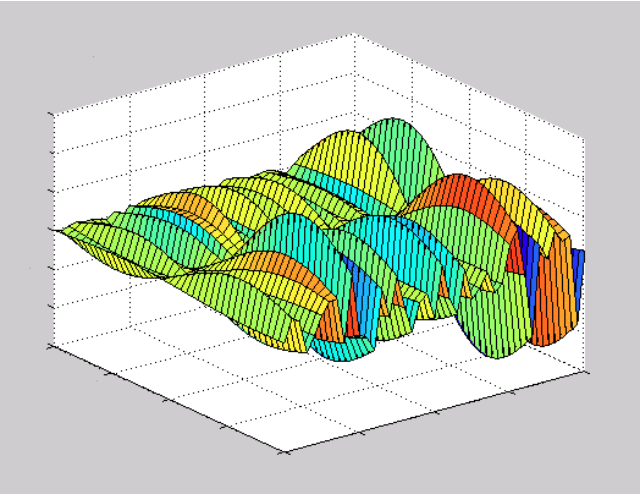


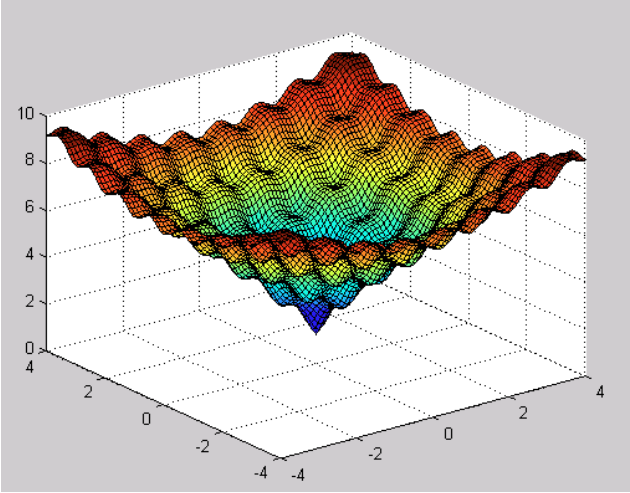
Geração 20

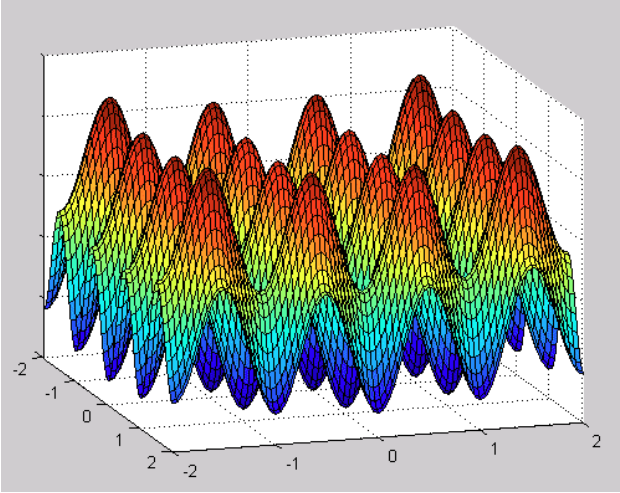
Espaço de Busca



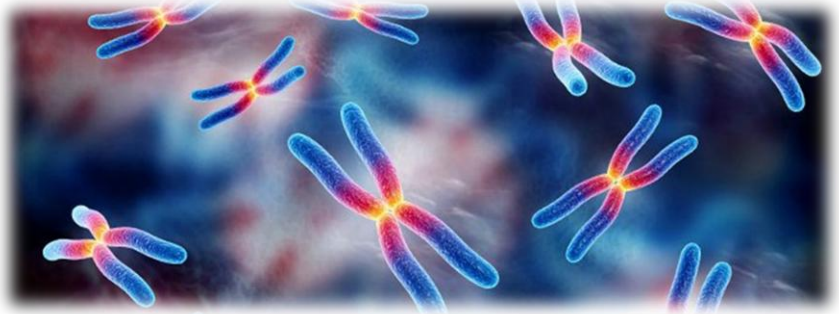




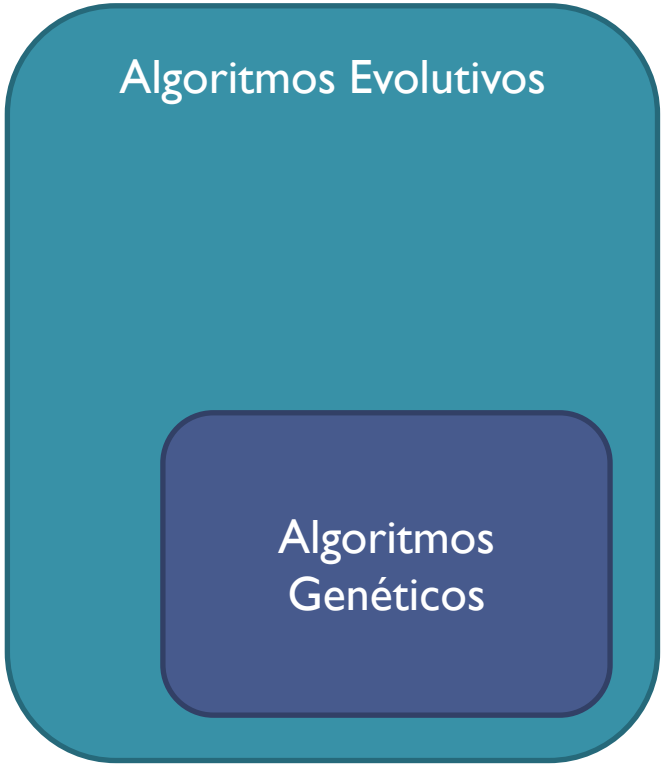




Algoritmos Genéticos

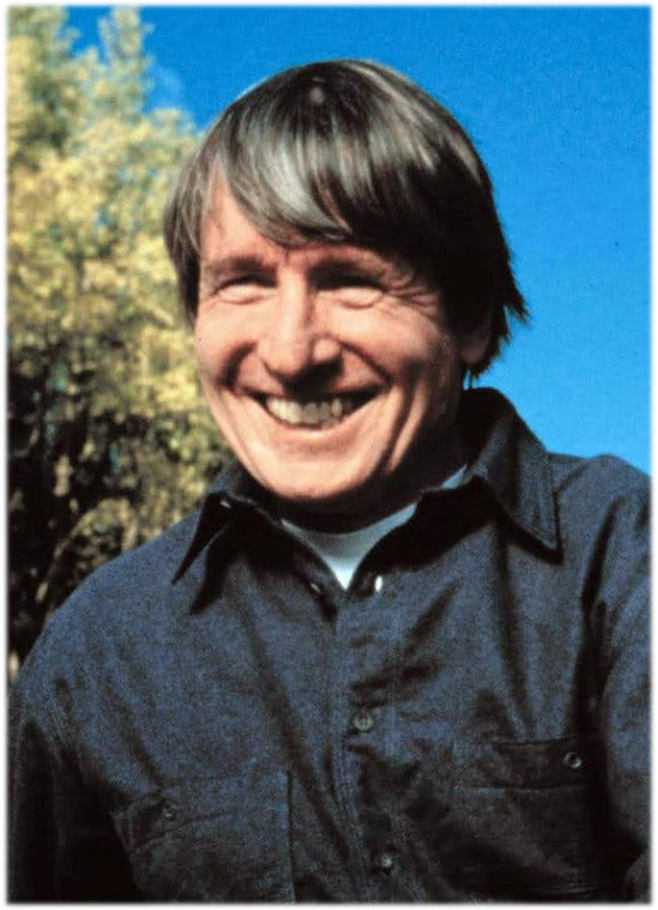


- Algoritmos Evolutivos em geral:
  - Baseados na Teoria da Evolução de Darwin.
- Algoritmos Genéticos:
  - Caso específico de Algoritmo Evolutivo.
  - Usam vocabulário emprestado da Genética Natural.



Algoritmos Genéticos

- Muitas versões diferentes:
  - As primeiras são dos anos 1950.
  - A versão de John Henry Holland, publicada em 1975, se tornou a mais popular.
  - Muitas outras variantes vem surgindo desde então, até os dias de hoje.

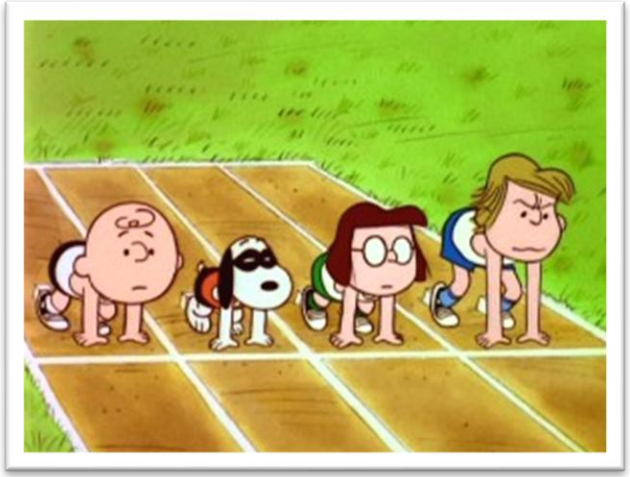


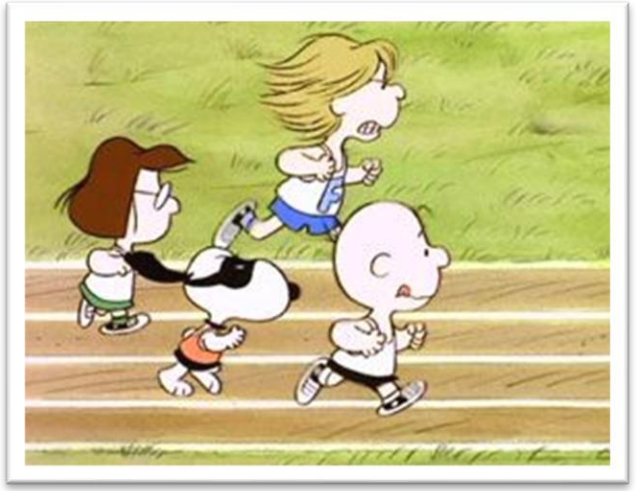
John Henry Holland [*1929 – †2015]

Holland, John Henry. "Adaption in natural and adaptive systems." (1975).

Algoritmos Genéticos

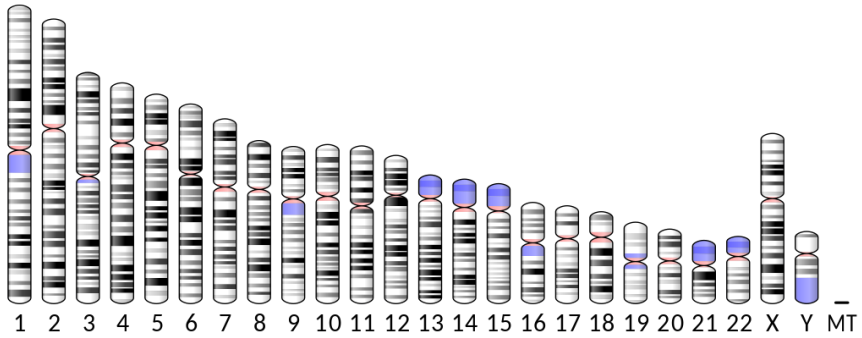
- Podem "evoluir" soluções para problemas do mundo real.
  - Problemas devem ser adequadamente codificados.
  - Deve haver uma forma de avaliar as soluções apresentadas.



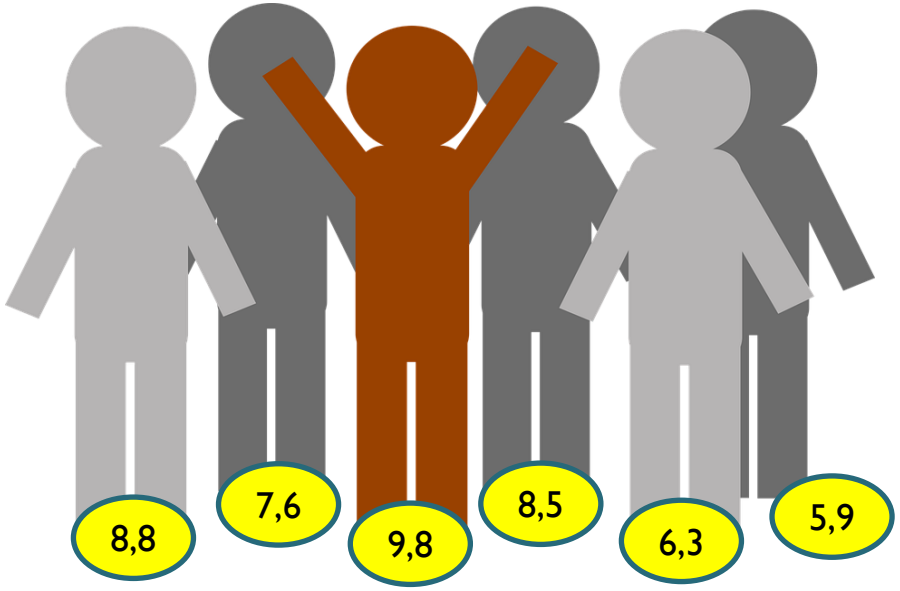


Indivíduo

- Possível solução para um dado problema.
  - Também chamado de cromossomo ou *string*.
- Codificado como vetor de características.
- A cada indivíduo é associado um valor de aptidão.
  - Mede qualidade da solução que ele representa.
- Conjunto de indivíduos forma uma população.



Esta Foto de Autor Desconhecido está licenciado em CC BY-SA



Exemplo: preparo de biscoitos

- Otimizar número de colheres de açúcar e farinha de trigo para preparar biscoitos.
- Passos:
  - Criar população inicial
  - Codificar *strings* ou cromossomos
  - Definir função de aptidão
  - Reprodução

Exemplo: preparo de biscoitos

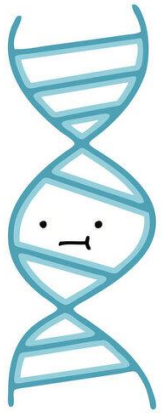
Farinha de trigo

0,3	0,6	0,5
0,7	0,9	0,6
0,6	0,7	0,4

Açúcar

Codificação

- Cada indivíduo é codificado por um conjunto de parâmetros (genes).
- Genes podem assumir valores:
  - Binários
    - String de binários {0,1} (*bitstrings*)
  - Inteiros
  - Reais
- Parâmetros são combinados para formar *strings* ou vetores (cromossomos).



Codificação

- Exemplo:
  - Cromossomo:

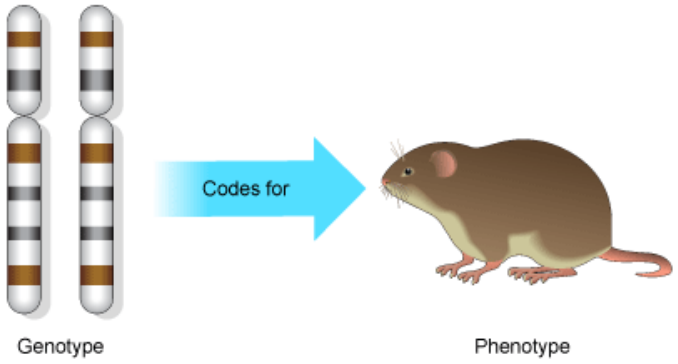
1	0	0	0	1	1	0	0	1	0

Locus

Gene

  - Alelos: {0,1}

Algoritmos Genéticos



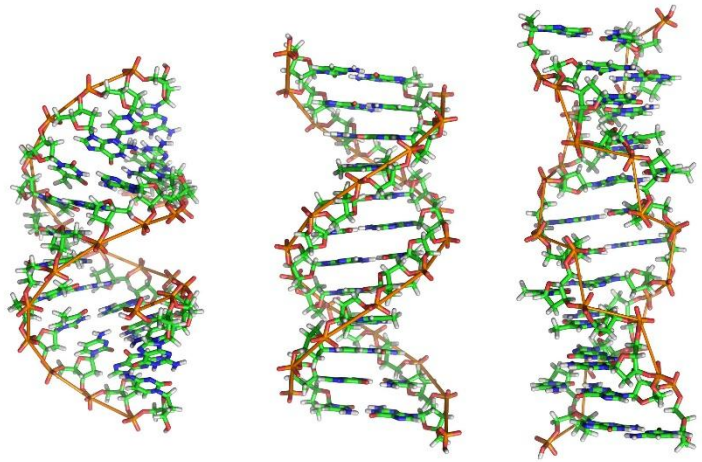
- Genótipo
  - Conjunto de parâmetros representado por um cromossomo.
  - Contém informação necessária para construir um organismo (fenótipo).
  - Fenótipo codificado.
- Fenótipo
  - Produto da interação de todos os genes.
  - Aptidão de um indivíduo depende do desempenho de seu fenótipo.

Representação

- Tradicionalmente, os indivíduos são representados por vetores binários:
  - 1 = presença
  - 0 = ausência
- Esta representação é independente do problema.
- Permite a utilização dos operadores de reprodução padrão.

Representação

- Genes também podem assumir valores inteiros, reais ou de tipos abstratos.
- Representações em níveis abstratos mais altos:
  - Facilitam sua utilização em determinados domínios mais complexos.
  - Necessitam de operadores específicos.



Esta Foto de Autor Desconhecido está licenciado em CC BY-SA

Representação

- Para alguns problemas, genes assumem valores inteiros ou reais.
  - Operadores específicos podem ser utilizados para esses valores.
  - Ou eles podem ser convertidos para valores binários.

$1x2^3$	$1x2^2$	$0x2^1$	$1x2^0$		$1x2^{-1}$	$0x2^{-2}$	$1x2^{-3}$	$1x2^{-4}$
1	**1**	**0**	**1**	**,**	**1**	**0**	**1**	**1**
8	4	0	1		0,5	0	0,125	0,0625

Ponto Binário

8 + 4 + 0 + 1 + 0,5 + 0,125 + 0,0625 = 13,6875 (Base 10)

Representação Binária

- Representar um valor $g \in \mathbb{Z}$ por um vetor de binários $b_1, b_2, \ldots, b_N$
  - Valor de g restrito ao intervalo $[g_{min}, \ldots, g_{max}]$ de valores inteiros.
  - Definir função $f[g] \rightarrow \{0,1\}^N$
  - Possibilidades:
    - Valor de g varia entre 0 a $2^N - 1$
    - Valor de g varia entre M a $M + 2^N - 1$
    - Valor de g varia entre 0 a $K - 1$, onde $K \neq 2^N - 1$
    - Etc...

Esta Foto de Autor Desconhecido
está licenciado em CC BY-SA

Representação Binária

- Valor de g varia entre 0 a $2^N - 1$
  - Converter valor de g para valor binário correspondente.
- Valor de g varia entre M a $M + 2^N - 1$
  - Converter valor de $g - M$ para valor binário correspondente.
- Desvantagem de ambos:
  - Valores próximos poderão ser totalmente diferentes em sua representação binária.
    - E vice-versa.

Esta Foto de Autor Desconhecido está licenciado em CC BY-NC-ND

Representação Binária

- Valor de g varia entre 0 a K, onde $K < 2^N - 1$
  - Duas abordagens podem ser seguidas:
    - Corte
    - Escala

Representação Binária

- Corte
  - Fácil de implementar.
  - Forte *bias* para os últimos valores.

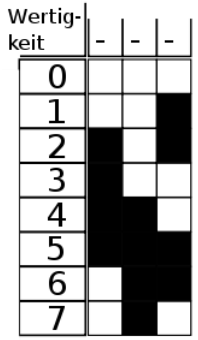
Valor de g	Código (x)
0	000
1	001
2	010
3	011
4	100
5	101
5	110
5	111

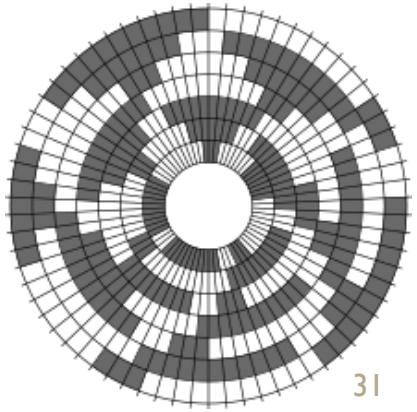
- Escala
  - Fácil de implementar.
  - Menor *bias*.
  - Maior custo computacional.

Valor de g	Código (x)
0	000
0	001
1	010
2	011
2	100
3	101
4	110
5	111

Código de Gray

- Criado por Frank Gray, físico e pesquisador do Bell Labs.
- Códigos para dígitos consecutivos diferem em apenas um bit.
- Projetado originalmente para evitar saídas falsas de chaves eletromecânicas.
- Hoje bastante usado para facilitar correção de erro em comunicação digital.
  - Exemplo: TV digital.





Esta Foto de Autor Desconhecido está licenciado em CC BY-SA
Esta Foto de Autor Desconhecido está licenciado em CC BY-SA

Exemplo

- Existem vários Códigos de Gray.
  - Não é único.
- Um Código de Gray para 3 bits:
  - 000, 010, 011, 001, 101, 111, 110, 100
- Um Código de Gray para 2 bits:
  - 00, 01, 11, 10

Inteiro	Binário	Código de Gray
0	0000	0000
1	0001	0001
2	0010	0011
3	0011	0010
4	0100	0110
5	0101	0111
6	0110	0101
7	0111	0100
8	1000	1100
9	1001	1101
10	1010	1111
11	1011	1110
12	1100	1010
13	1101	1011
14	1110	1001
15	1111	1000

Código de Gray

- Algoritmo Simples:

1. Começar com todos os bits iguais a zero.
2. Para cada novo número:
   Mudar o valor do bit mais à direita que gerar uma nova sequência de bits.

Código Termômetro

- Valores consecutivos diferem em apenas um bit.
  - Utiliza muitos dígitos binários.
  - Tamanho cresce linearmente com número de valores.

Inteiro	Binário	Termômetro
0	0000	0000
1	0001	0001
2	0010	0011
3	0011	0111
4	0100	1111

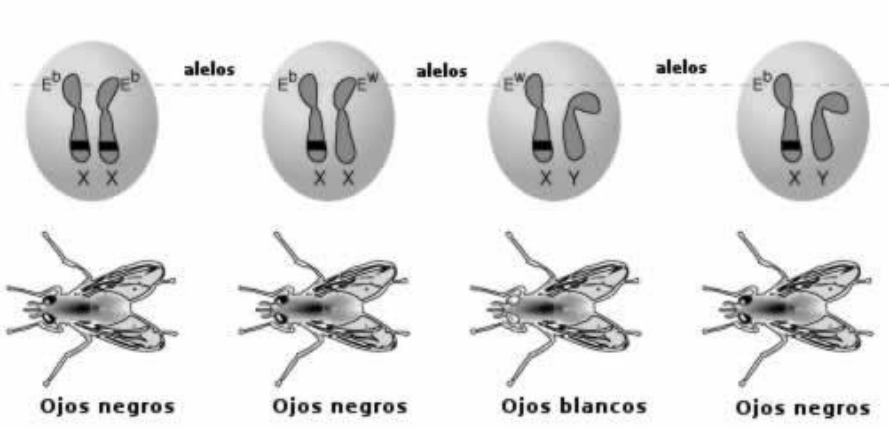
Esta Foto de Autor Desconhecido está licenciado em CC BY-SA-NC

Função de Aptidão

- Mede o grau de aptidão de um indivíduo.
  - Retorna um valor (índice) de aptidão numérico.
    - Proporcional à utilidade ou habilidade do indivíduo.
  - Aptidão = probabilidade do indivíduo sobreviver para a próxima geração.
    - Cada aplicação tem sua própria função de aptidão.

Função de Aptidão

- É aplicada ao fenótipo do indivíduo.
  - Genótipo precisa ser decodificado, recuperando o fenótipo associado.
- Pode envolver uma (otimização de função) ou mais medidas (otimização multi-objetivo).
  - Ex. projeto de ponte
    - Custo, tempo de construção, capacidade máxima.



Esta Foto de Autor Desconhecido está licenciado em CC BY-SA

Sydney Harbour Bridge – Sydney, Australia por Fabricio Breve

Função de Aptidão

- Métodos para calcular valor ou índice de aptidão:
  - Padrão
  - Baseada em ranking
  - Ranking-espaço

Esta Foto de Autor Desconhecido está licenciado em CC BY-SA

Função de Aptidão Padrão

- Utiliza apenas informação sobre "qualidade do cromossomo".

$$f_i = \frac{q_i}{\sum_j q_j}$$

q = índice de aptidão do cromossomo

Função de Aptidão Padrão

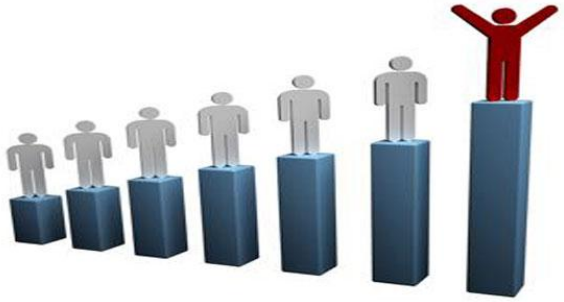
- Exemplo:
  - Preparo de biscoitos
    - Codificação: [colheres-farinha colheres-açúcar]

$$f_i = \frac{q_i}{\sum_j q_j}$$

Esta Foto de Autor Desconhecido está licenciado em CC BY-SA-NC

Cromossomo	q_i	f_i
[1 3]	4	0,20
[3 1]	3	0,15
[1 2]	7	0,35
[1 1]	6	0,30

Aptidão Baseada em Ranking



- Aptidão Padrão:
  - Uma escolha infeliz da escala do índice de aptidão pode prejudicar reprodução.
    - Ex.: 1, 10, 100
- **Aptidão baseada em ranking** utiliza medida de qualidade apenas para definir um "ranking" de cromossomos por aptidão.
  - Utiliza depois um algoritmo para selecionar candidato por ranking.

Esta Foto de Autor Desconhecido está licenciado em CC BY-SA-NC

Aptidão Baseada em Ranking

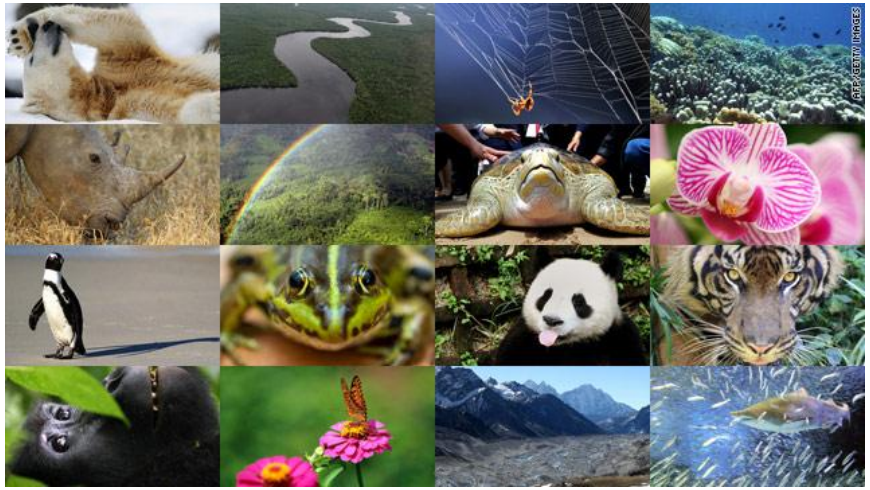
- Exemplo:
  - Sejam os valores de aptidão do ranking: 0,60; 0,25; 0,10; e 0,05.

Esta Foto de Autor Desconhecido está licenciado em CC BY-NC-ND

Cromossomo	q_i	Ranking	f_i
[1 4]	4	3	0,10
[3 1]	3	4	0,05
[1 2]	7	1	0,60
[1 1]	6	2	0,25

Aptidão Ranking-Espaço

- Medidas de aptidão anteriores ignoram diversidade.
  - Diversidade mede o quão diferente são os cromossomos de uma população.
- Princípio da diversidade:
  - Ser diferente pode ser tão bom quanto ser apto.



Esta Foto de Autor Desconhecido está licenciado em CC BY-SA-NC

Aptidão Ranking-Espaço

- Medida de Diversidade:
  - Utiliza ranking de medidas:

$$D_i = \sum_j d_{ij}^2$$

d_{ij} = medida de distância entre i e j

  - Aptidão Ranking-Espaço:

$$Rank_i = \alpha.Rank_{f_i} + \beta.Rank_{D_i}$$

α e β usados para ponderar importância de diversidade e aptidão, respectivamente

Aptidão Ranking-Espaço

1. Selecionar indivíduo com maior aptidão baseada em ranking.
2. Enquanto a população não estiver completa:

   Selecionar indivíduo com maior aptidão ranking-espaço /* diversidade é medida com relação aos indivíduos já selecionados */

   Em caso de empate, desempatar por ranking ou diversidade.

Seleção

- Escolhe preferencialmente indivíduos com maiores notas de aptidão.
  - Não exclusivamente.
    - Procura manter a diversidade da população.
- Indivíduos mais aptos têm mais oportunidades de gerar descendentes.
  - Que serão cada vez mais aptos.

Seleção

- Seleciona uma população intermediária.
  - Onde serão aplicados os operadores de reprodução.
- Existem vários métodos de seleção:
  - Por Roleta
  - Por Torneio
  - Amostragem Universal Estocástica

Seleção pela Roleta

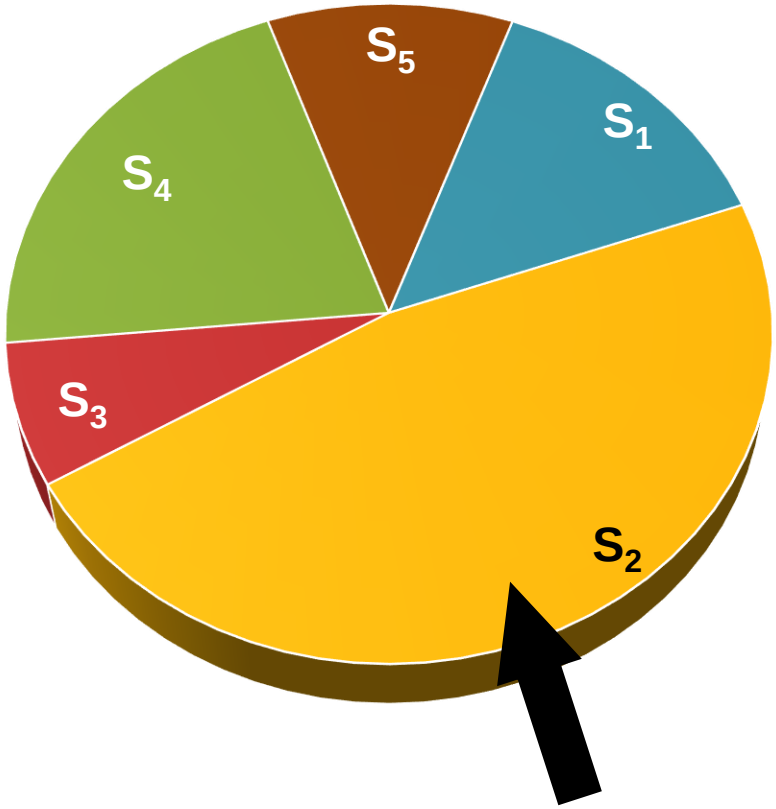
- Método mais simples e utilizado.
- Escolhe indivíduos para fazer parte da próxima geração por meio de um sorteio.
- Cada indivíduo da população é representado na roleta por uma fatia proporcional ao seu índice de aptidão.
  - Quanto maior o desempenho, maior é a chance de ser selecionado para a próxima geração.

Esta Foto de Autor Desconhecido está licenciado em CC BY-SA-NC

Seleção pela Roleta

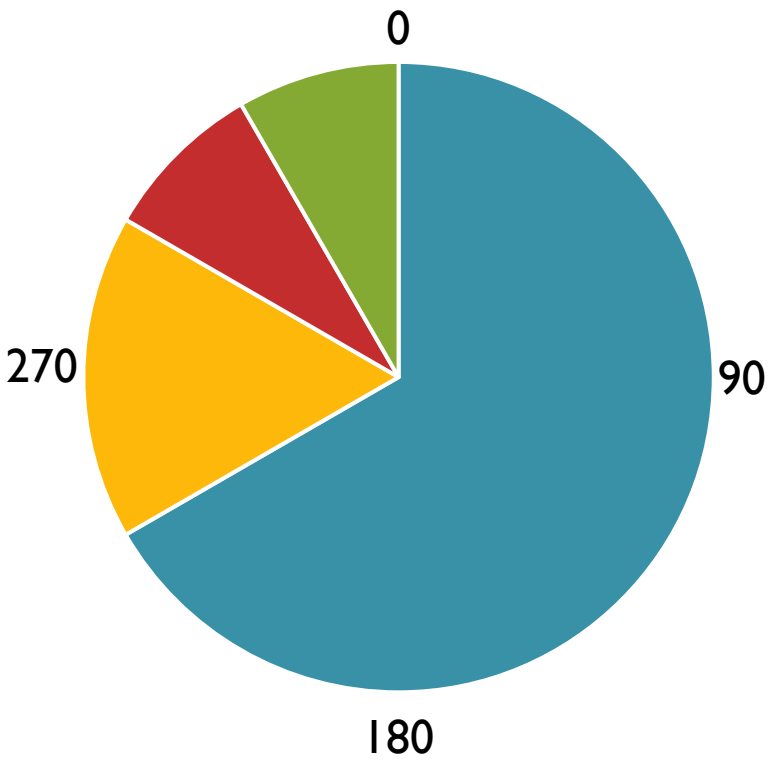
- Método da Roleta baseado em Aptidão Relativa

Indivíduo (S_i)	Aptidão $f(S_i)$	Aptidão Relativa
S_1 10110	2,23	0,14
S_2 11000	7,27	0,47
S_3 11110	1,05	0,07
S_4 01001	3,35	0,21
S_5 00110	1,69	0,11



Seleção pela Roleta

Ind.	Cromossomo	Aptidão	Graus
1	0001100101010	16	240
2	0101001010101	4	60
3	1011110100101	2	30
4	1010010101001	2	30



Seleção por Torneio

- Escolha n indivíduos da população aleatoriamente, com a mesma probabilidade.
  - Geralmente $n = 3$
- Cromossomo com maior aptidão dentre estes n cromossomos é selecionado para a população intermediária.
- Processo se repete até que a população intermediária seja preenchida.

Seleção por Torneio

- Método de Torneio baseado em Aptidão Relativa:

Indivíduo (S_i)	Aptidão $f(S_i)$	Aptidão Relativa
S_1 10110	2,23	0,14
S_2 11000	7,27	0,47
S_3 11110	1,05	0,07
S_4 01001	3,35	0,21
S_5 00110	1,69	0,11

Com $n = 3$

Candidatos			Vencedor
S_1	S_2	S_5	S_2
S_2	S_4	S_5	S_2
S_5	S_1	S_3	S_1
S_4	S_5	S_3	S_4
S_3	S_1	S_5	S_1

Seleção por Torneio

- Possui um parâmetro que permite definir explicitamente a pressão seletiva durante a evolução:
  - Tamanho do torneio:
    - Quanto maior o número de indivíduos que participam do torneio, maior a pressão seletiva.
      - Indivíduo tem que ser melhor que uma quantidade maior de competidores.

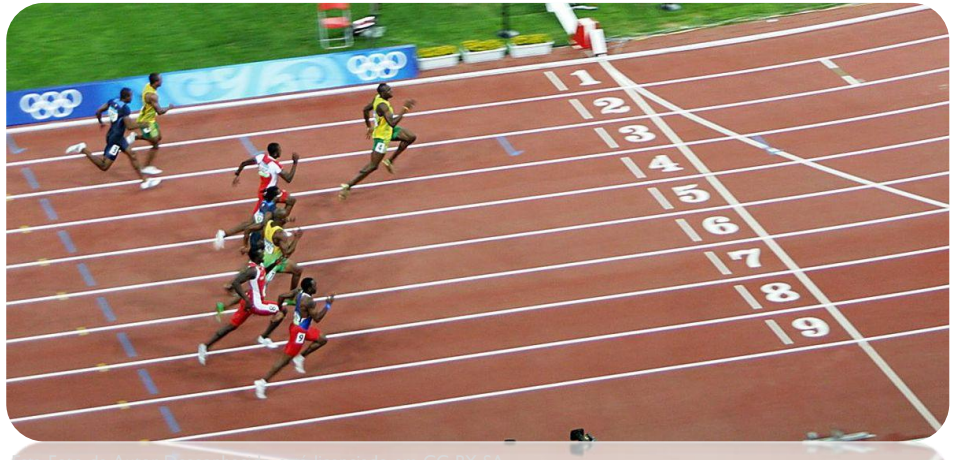


Foto de Autor Desconhecido está licenciado em CC BY-SA

11396
8 290
ACTIVISION

SKOOTE CASIO
AERIAL TURN
0:15
16650
500
HOLLYWOOD
SANTA CRUZ
OX
JAM

01"45 FROM 2ND
BEST LAP
0'37"26
3 LAPS
2'05"39
TIME
0'20"20
0'20"20
0'00"00
0'00"00
POSITION:
1 ST
LAP 1/3
319km/h

JALECO
JALECO
01
BRA
00
ITA

1P 136700 HI 136700 PRESS
KO
CHUN-LI 97
KEN

BRA 02 URU 01
02:27

Pressão Seletiva

- Grau com que os melhores indivíduos são favorecidos.
  - Influencia taxa de convergência do algoritmo genético.
    - Pressão muito baixa:
      - Taxa de convergência lenta.
      - Demora para encontrar boa solução.
    - Pressão muito elevada:
      - Convergência prematura.

Esta Foto de Autor Desconhecido está licenciado em CC BY-NC-ND

Diversidade

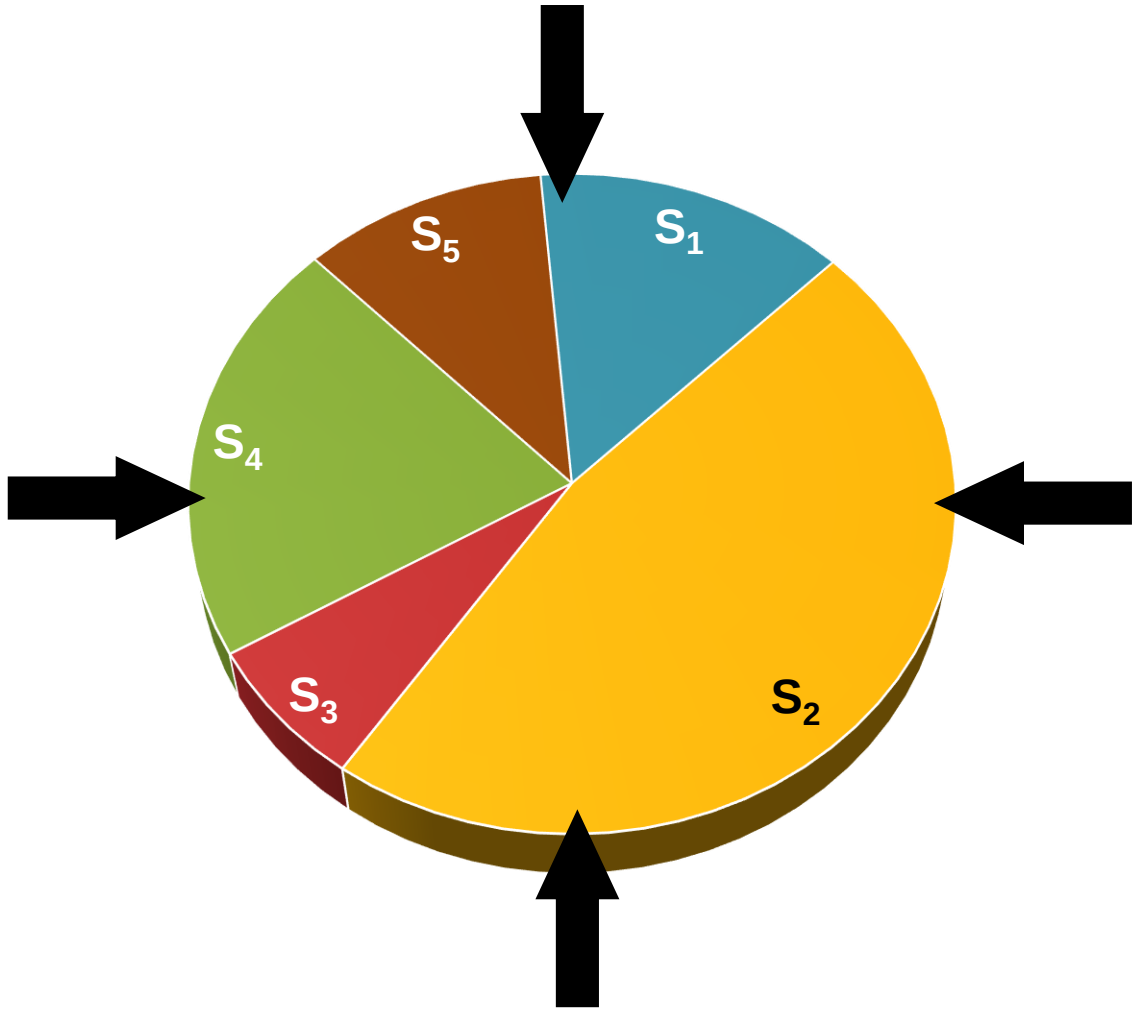
- Deve haver equilíbrio entre pressão seletiva e diversidade.
- Formas de prevenir convergência prematura:
  - Controlar número de oportunidades de reprodução de cada indivíduo.
- Formas de promover diversidade:
  - Aumento do tamanho da população.
  - Aumento da taxa de mutação.

https://www.deviantart.com/krdoz/art/Biker-Gang-216878091

Seleção por Amostragem Universal Estocástica

- Conhecido como SUS (do inglês, *Stochastic Universal Sampling*).
- Variação do método da roleta.
  - Utiliza n agulhas igualmente espaçadas em vez de apenas 1.
    - Onde n é o número de indivíduos a serem selecionados para a próxima geração.
    - Em vez de n vezes, a roleta é girada uma única vez.
  - Exibe menos variância que as repetidas chamadas do método da roleta.

Seleção por Amostragem Universal Estocástica

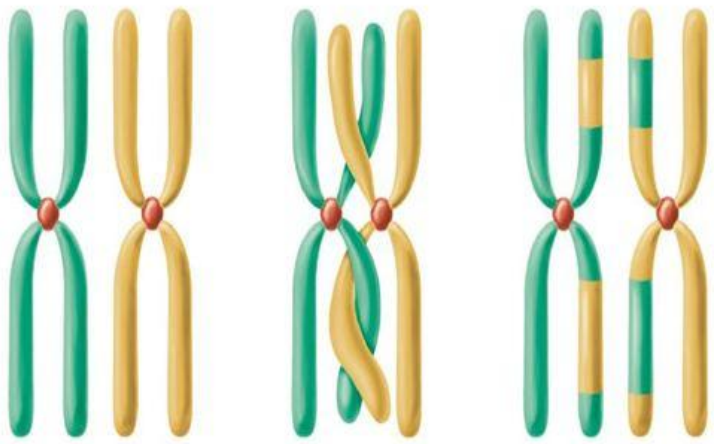


Seleção por Estado Estável

- Maioria dos algoritmos genéticos é geracional.
  - A cada geração, a população é formada apenas pelos filhos da geração anterior.
  - Estado estável (*steady state*) reduz o número de pais que serão substituídos pelos filhos.
    - Existe uma intersecção entre conjuntos de indivíduos de gerações sucessivas.
      - Ex.: pegar os N/2 melhores entre os N pais e N filhos.

Operadores Genéticos

- Permitem obtenção de novos indivíduos.
  - Cada geração possui, geralmente, indivíduos mais aptos.
  - Principais operadores genéticos:
    - *Crossover* (cruzamento ou recombinação)
    - Mutação
    - Elitismo



Crossover

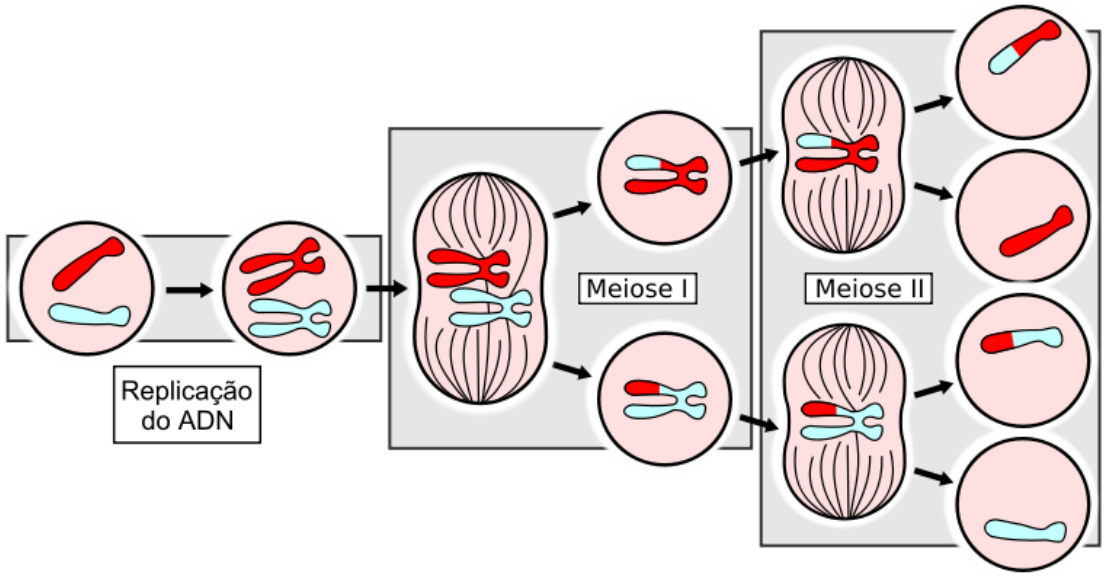
- Filhos herdam partes das características dos pais durante a reprodução.
  - Permite que as próximas gerações herdem estas características.
- Funcionamento:
  - Escolhe dois indivíduos e troca trechos entre eles.

Esta Foto de Autor Desconhecido está licenciado em CC BY-SA-NC

Esta Foto de Autor Desconhecido está licenciado em CC BY-SA

Crossover

- Exemplo:



Ponto de Crossover, $(r = 5)$

1	0	0	0	0	0	1	1

Cromossomo Pai 1

1	1	0	1	1	1	1	0

Cromossomo Pai 2

Crossover de

Ponto Único

1	0	0	0	1	1	1	0

Cromossomo Filho A

1	1	0	1	0	0	1	1

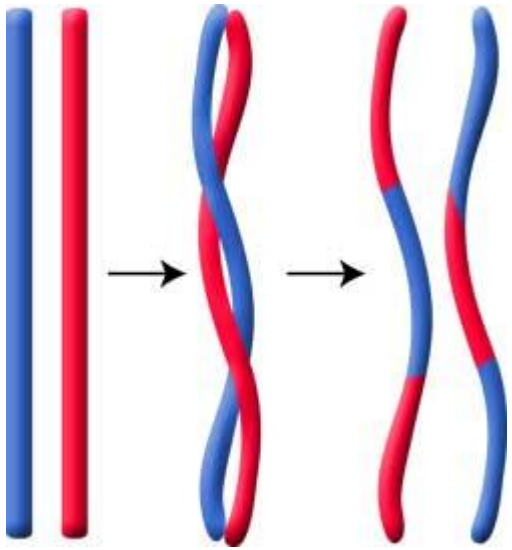
Cromossomo Filho B

Crossover

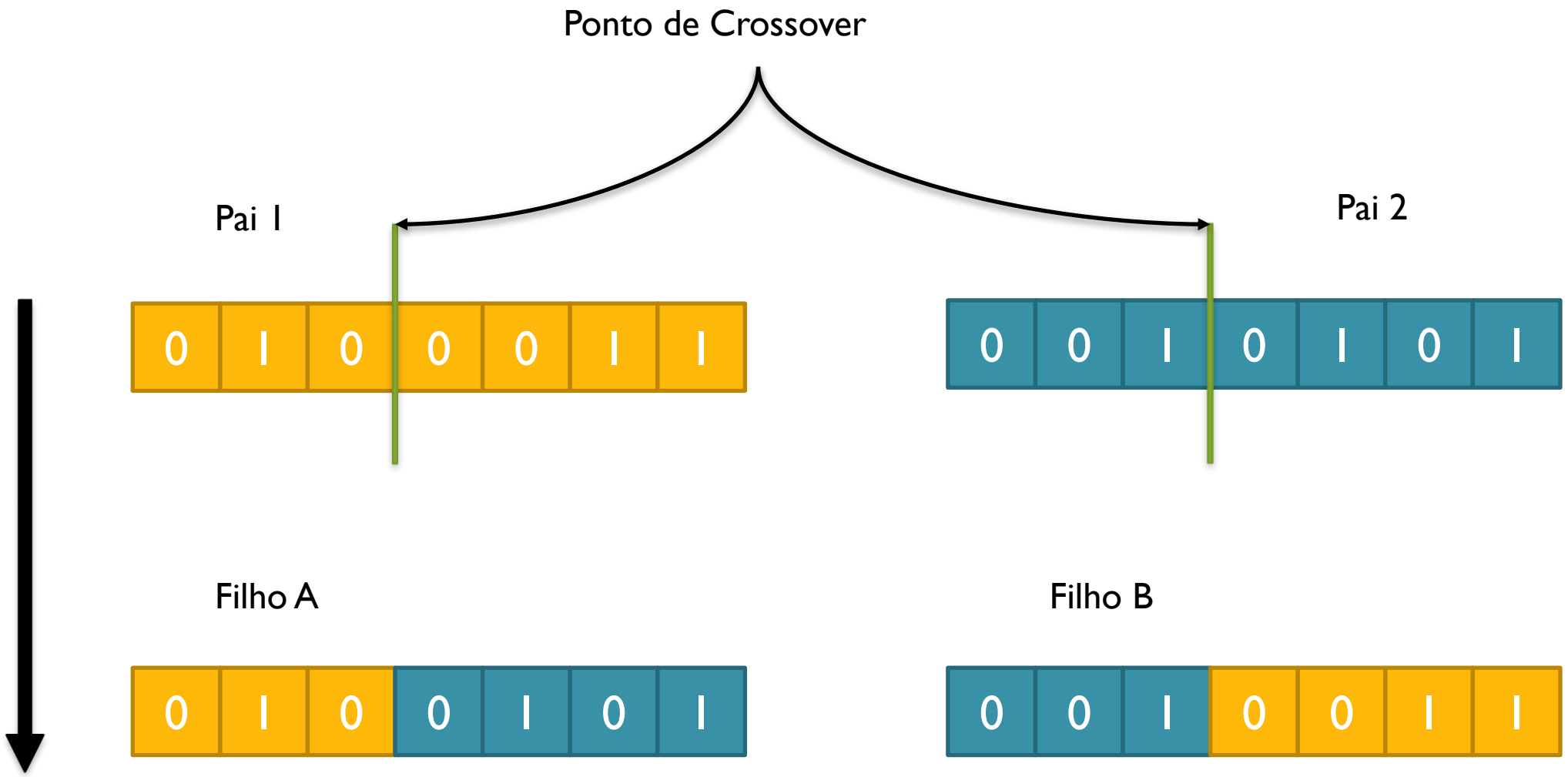
- É o operador genético predominante.
  - A taxa de *crossover* deve ser maior que a taxa de mutação.
  - Taxa de *crossover*: $0{,}6 < P_C < 1{,}0$
  - Caso *crossover* não seja aplicado, descendentes são iguais aos pais.
- É a operação mais importante para exploração rápida do espaço de busca.

Crossover

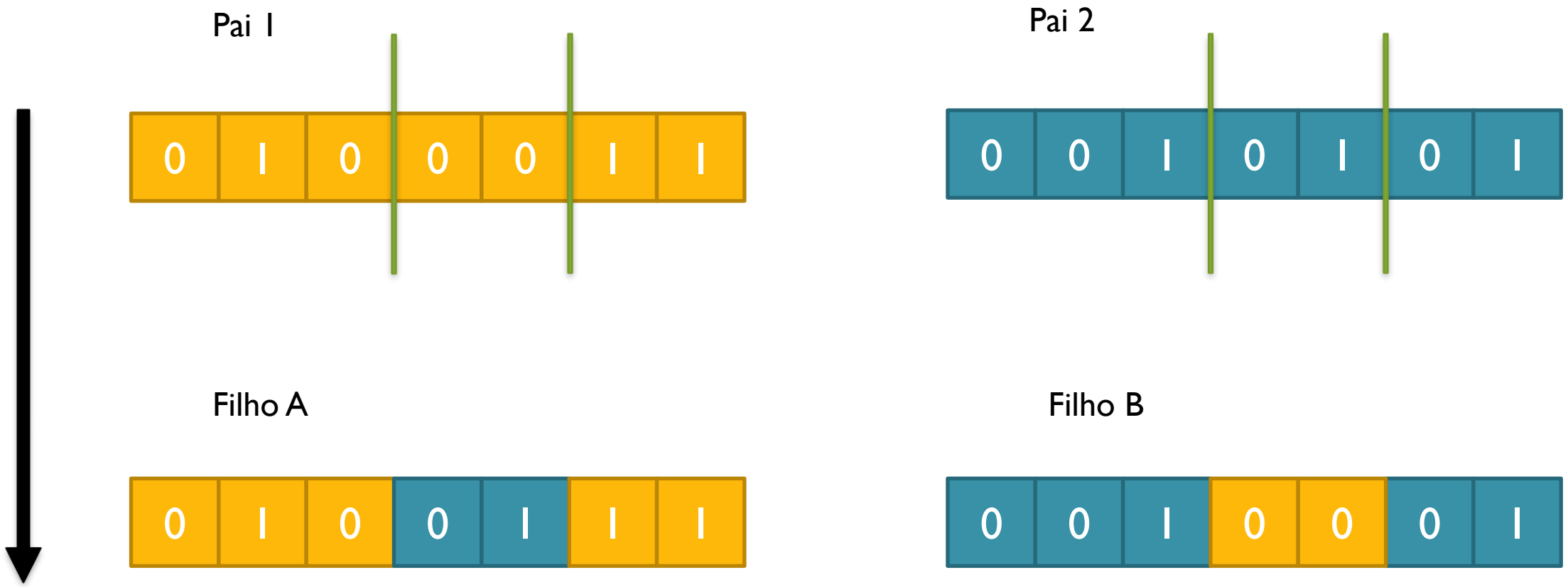
- Diversas variações:
  - Um ponto
    - Mais comum
  - Dois pontos
  - Multi-pontos
  - Uniforme



Crossover de Um Ponto



Crossover de Dois Pontos



Crossover Uniforme

Máscara:

0	1	0	1	0	0	0

Pai 1

0	1	0	0	0	1	1

Pai 2

0	0	1	0	1	0	1

Filho A

0	0	0	0	0	1	1

Filho B

0	1	1	0	1	0	1

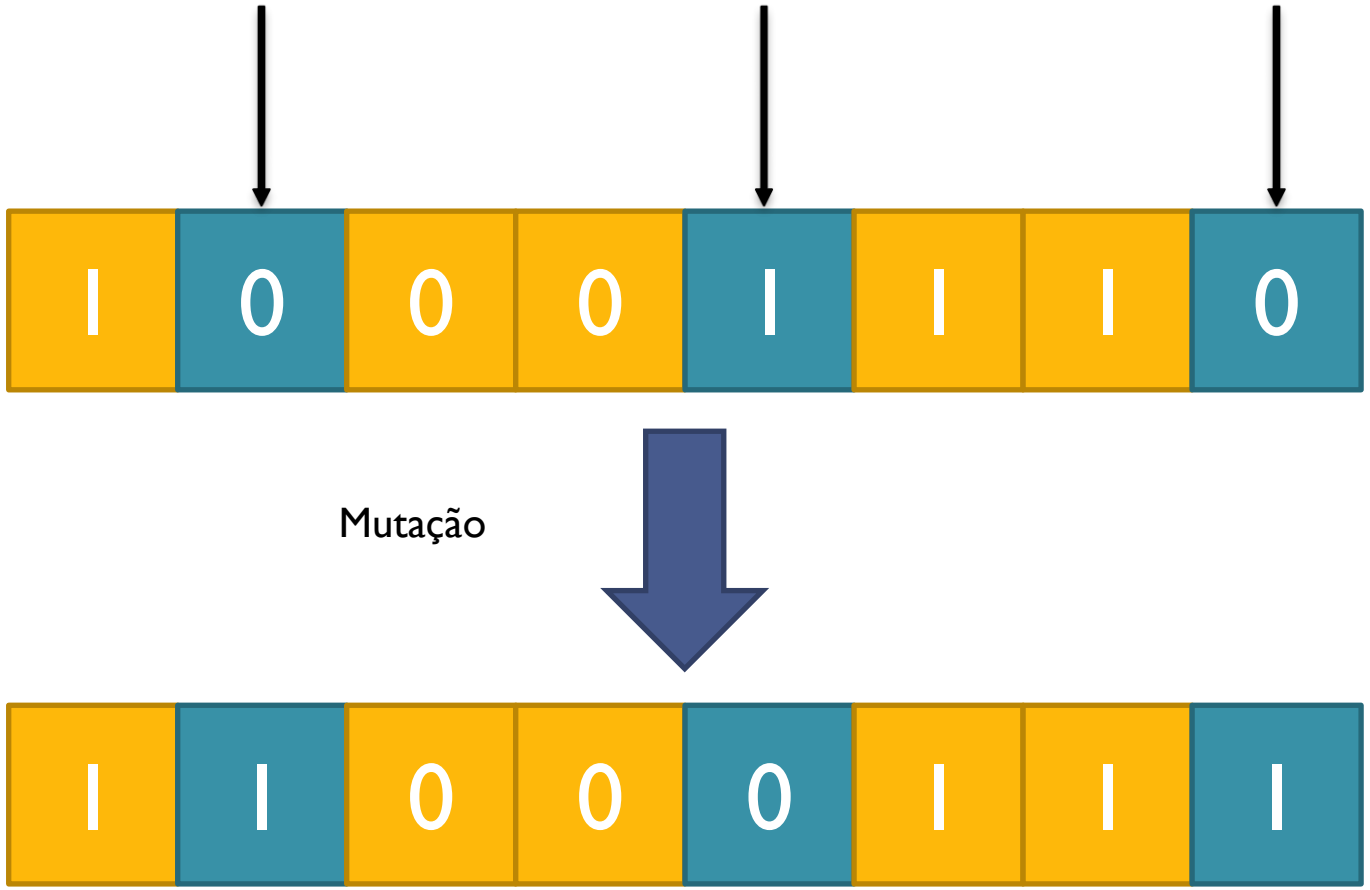
Mutação

- Permite introdução e manutenção da diversidade genética.
  - Aplicado a cada indivíduo após *crossover*.
- Funcionamento:
  - Altera aleatoriamente um ou mais componentes de uma estrutura escolhida.

Esta Foto de Autor Desconhecido está licenciado em CC BY-SA

Mutação

- Exemplo



Bits a serem mutados

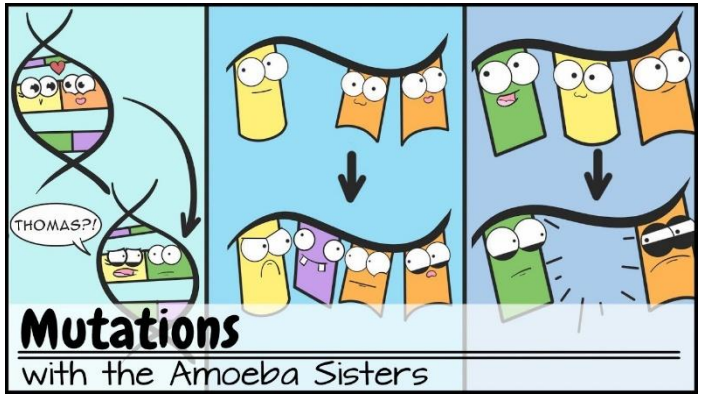
Cromossomo Original

Cromossomo Mutado

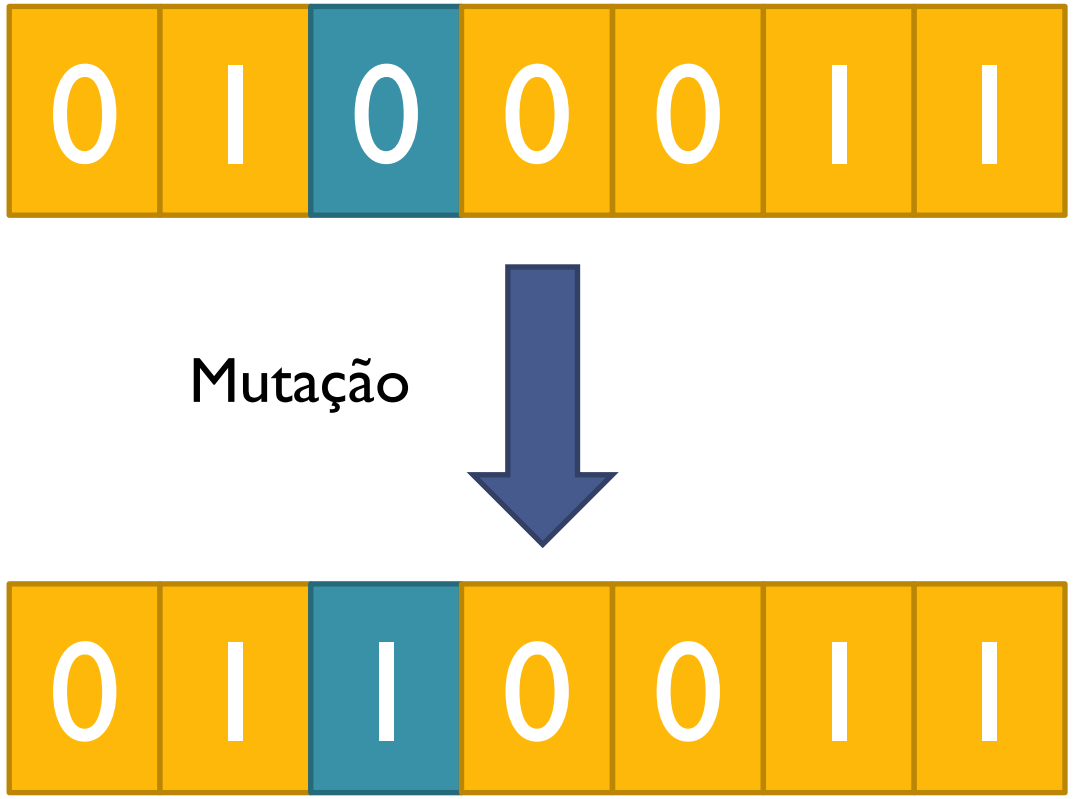
Mutação

- Assegura que a probabilidade de atingir qualquer ponto do espaço de busca nunca será zero.
  - Reduz chance de parada em Mínimos Locais.
- Operador genético secundário.
  - Taxa de mutação pequena $P_M \approx 0{,}001$

Mutação



- Mais um exemplo:



Cromossomo Original

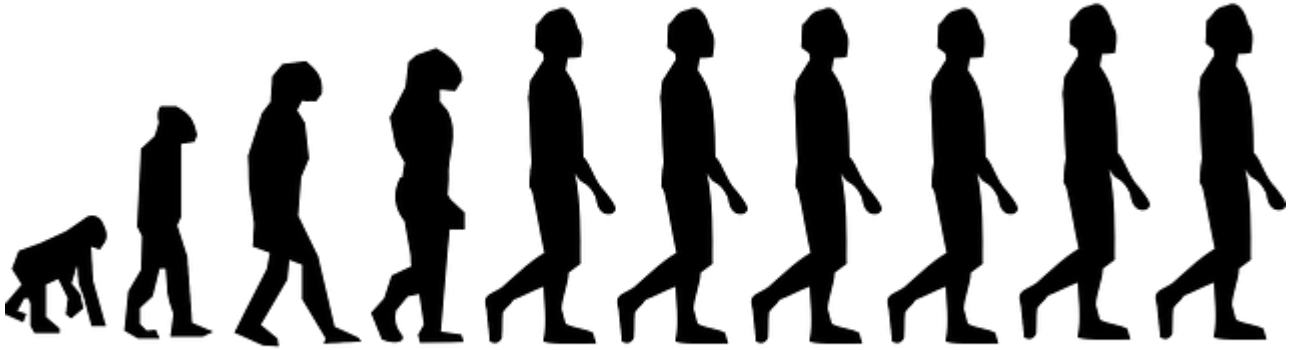
Cromossomo Mutado

Elitismo

- Indivíduo de maior desempenho é automaticamente selecionado.
- Evita modificações deste indivíduo pelos operadores genéticos.
  - Utilizado para que os melhores indivíduos não desapareçam da população.

Funcionamento do Algoritmo Genético

- Se o algoritmo genético estiver corretamente implementado, a população geralmente evolui em gerações sucessivas.
  - Até estabilizar.
- Aptidões do melhor indivíduo e do indivíduo médio aumentam em direção a um ótimo global.

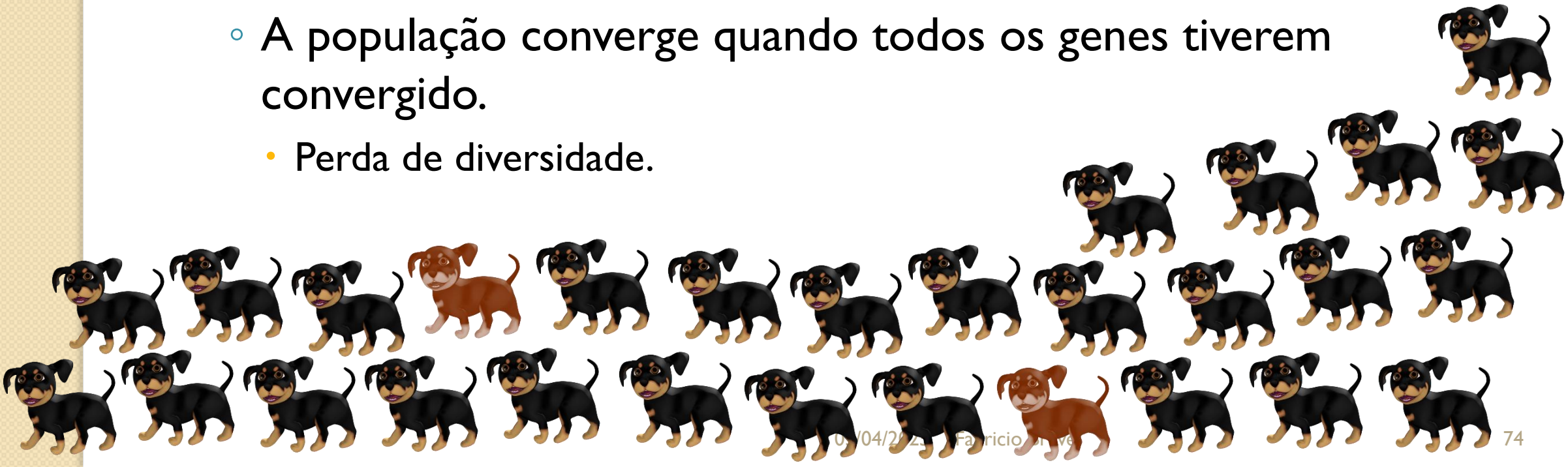


Critério de Parada

- Tempo de execução.
- Número de gerações.
- Valor de aptidão mínimo, médio e/ou máximo.
- Convergência:
  - Nas últimas k iterações não houve melhora nas aptidões.

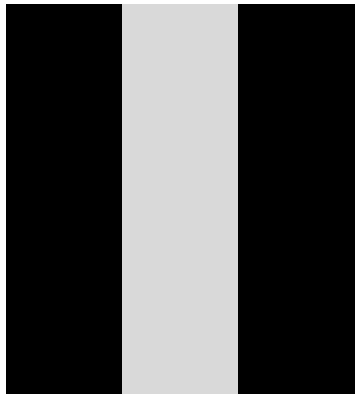
Convergência

- Convergência é a progressão em direção a uma uniformidade crescente.
  - Um gene converge quando 95% da população compartilha o mesmo valor.
  - A população converge quando todos os genes tiverem convergido.
    - Perda de diversidade.



Exemplo: Reconhecimento de Padrões

- Considere um problema de evoluir uma população para reconhecer um caractere, no caso o número '1'.
  - 4 linhas e 3 colunas



(a)

$$\begin{bmatrix} 0 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \end{bmatrix}$$

(b)

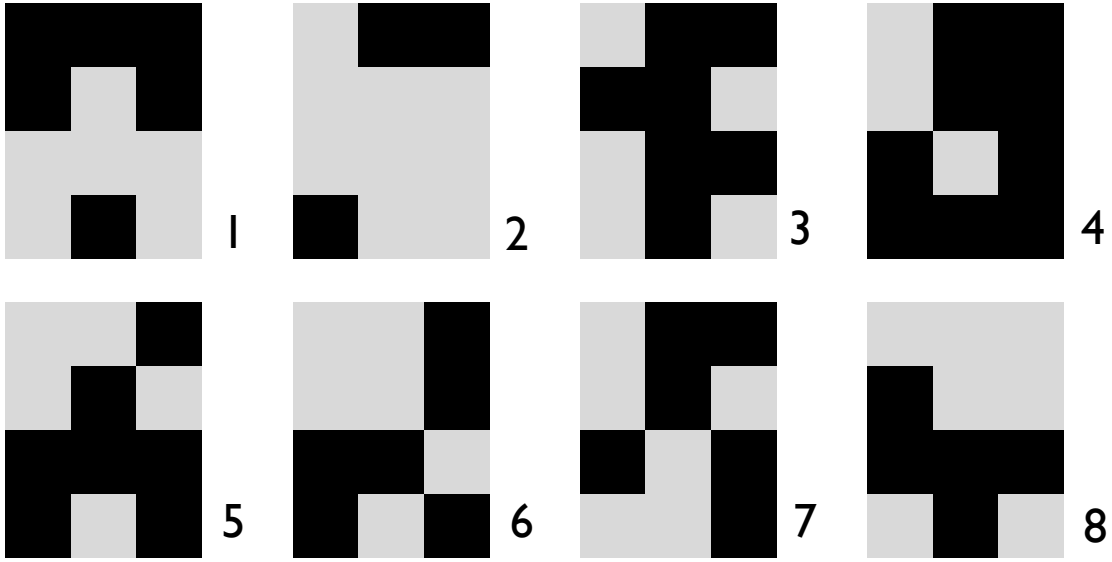
Caractere a ser reconhecido usando um algoritmo genético padrão. (a) Representação Pictórica (b) Representação em Matriz

Exemplo: Reconhecimento de Padrões

- Utilizaremos uma população de 8 indivíduos.
- Cada indivíduo será representado por um vetor, que corresponde às 4 linhas da matriz colocadas uma após a outra.
- Exemplo:
  - $\boldsymbol{x}_1 = [010010010010]$

Exemplo: Reconhecimento de Padrões – Inicialização Aleatória

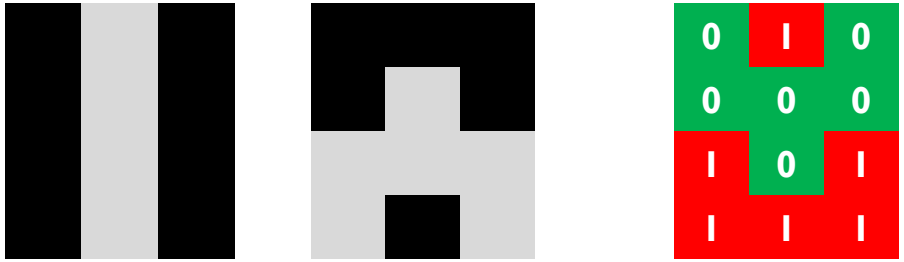
$$P = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 0 \\ 1 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 1 \end{bmatrix}$$



Avaliação de Aptidão

- Maneira mais simples de determinar aptidão:
  - Contar o número de bits diferentes entre cada indivíduo da população e o indivíduo alvo.
    - Chamado *Distância de Hamming*.
    - Exemplo:
      - Dada a população inicial, o vetor com as Distâncias de Hamming de cada indivíduo para o alvo é:
        - $\boldsymbol{h} = [6, 7, 9, 5, 5, 4, 6, 7]$

Exemplo:



Alvo Indivíduo 1 Distância

Avaliação de Aptidão

- Aptidão da população:
  - Obtida tomando o tamanho $l = 12$ de cada indivíduo e subtraindo a distância de Hamming obtida.
  - Exemplo (na nossa população inicial):
    - Considerando $\boldsymbol{h} = [6, 7, 9, 5, 5, 4, 6, 7]$
    - $\boldsymbol{f} = [f_1, \ldots, f_8] = [6, 5, 3, 7, 7, 8, 6, 5]$
- Aptidão do indivíduo ideal:
  - $f = 12$
- Critério de Parada:
  - Ao menos um indivíduo tem aptidão $f = 12$

Seleção por Roleta

- Some a aptidão de todos os indivíduos:

$$f_T = \sum_i f_i \ , \quad i = 1, \ldots, N$$

- Multiplique a aptidão de cada indivíduo por 360 e divida pela soma da aptidão, determinando a porção da roleta atribuída a cada indivíduo:

$$f_i^{'} = (360.f_i) / f_T \ , \quad i = 1, \ldots, N$$

- Gere um número aleatório no intervalo (0,360] e compare-o com o intervalo pertencente a cada indivíduo:
  - Selecione o indivíduo cujo intervalo contém o número sorteado e o coloque na nova população **P**

Seleção por Roleta

Indivíduo	Cromossomo	Aptidão	Graus	Porção da Roleta
1	000010111101	6	46	(0,46]
2	100111111011	5	38	(46,84]
3	100001100101	3	23	(84,107]
4	100100010000	7	54	(107,161]
5	110101000010	7	54	(161,215]
6	110110001010	8	61	(215,276]
7	100101010110	6	46	(276,322]
8	111011000101	5	38	(322,360]

Obs: valores arredondados com propósitos didáticos

Seleção por Roleta

- Assuma que os valores sorteados aleatoriamente foram:
  - $\boldsymbol{s} = [230, 46, 175, 325, 275, 300, 74, 108]$
- Portanto a população $\mathbf{P}$ será composta por:
  - $\mathbf{P} = [\boldsymbol{x}_6, \boldsymbol{x}_1, \boldsymbol{x}_5, \boldsymbol{x}_8, \boldsymbol{x}_6, \boldsymbol{x}_7, \boldsymbol{x}_2, \boldsymbol{x}_4]^T$
- Observe:
  - $\boldsymbol{x}_3$ foi extinto.
  - $\boldsymbol{x}_6$ foi selecionado duas vezes.

Reprodução

- Temos 4 pares de indivíduos:
  - $\boldsymbol{x}_6$ e $\boldsymbol{x}_1$
  - $\boldsymbol{x}_5$ e $\boldsymbol{x}_8$
  - $\boldsymbol{x}_6$ e $\boldsymbol{x}_7$
  - $\boldsymbol{x}_2$ e $\boldsymbol{x}_4$
- Para cada par geramos um número aleatório r no intervalo $[0,1]$.
  - Se $r \leq P_c$, então o cruzamento é feito.
    - Lembre-se: P_c é a taxa de *crossover*.
  - Senão, os pais originais são repetidos na próxima geração.

Reprodução

- Seja $\boldsymbol{r} = [0{,}5;\ 0{,}7;\ 0{,}3;\ 0{,}9]$ os números aleatórios gerados e $P_C = 0{,}6$
  - Portanto o primeiro e o terceiro par foram selecionados para *crossover*:
    - $\boldsymbol{x}_6$ e $\boldsymbol{x}_1$
    - $\boldsymbol{x}_6$ e $\boldsymbol{x}_7$
- Deve ser definido o *ponto de crossover* (cp) para cada caso.
  - Definidos aleatoriamente.
    - Assuma que foram sorteados $cp = 5$ e $cp = 9$ para cada um dos pares respectivamente.

Reprodução

Ponto de crossover, $cp = 5$

110110001010 Cromossomo pai 1 (x_6)
000010111101 Cromossomo pai 2 (x_1)

110110111101 Cromossomo filho A
000010001010 Cromossomo filho B

Ponto de crossover, $cp = 9$

110110001010 Cromossomo pai 1 (x_6)
100101010110 Cromossomo pai 2 (x_7)

110110001110 Cromossomo filho A
000011010010 Cromossomo filho B

Reprodução

- Nova população intermediária **P** é:

$$P = \begin{bmatrix} 1 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Mutação

- Assuma uma probabilidade de mutação $P_M = 0{,}02$
  - Para cada posição na matriz um número aleatório *r* é gerado e comparado com P_M
    - Se $r \leq P_M$
      - Bit correspondente é selecionado para ser mutado.

Mutação

$$P = \begin{bmatrix} 1 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & \mathbf{0} & 1 & 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 & 1 & 1 & \mathbf{1} & 1 & 1 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix}$$

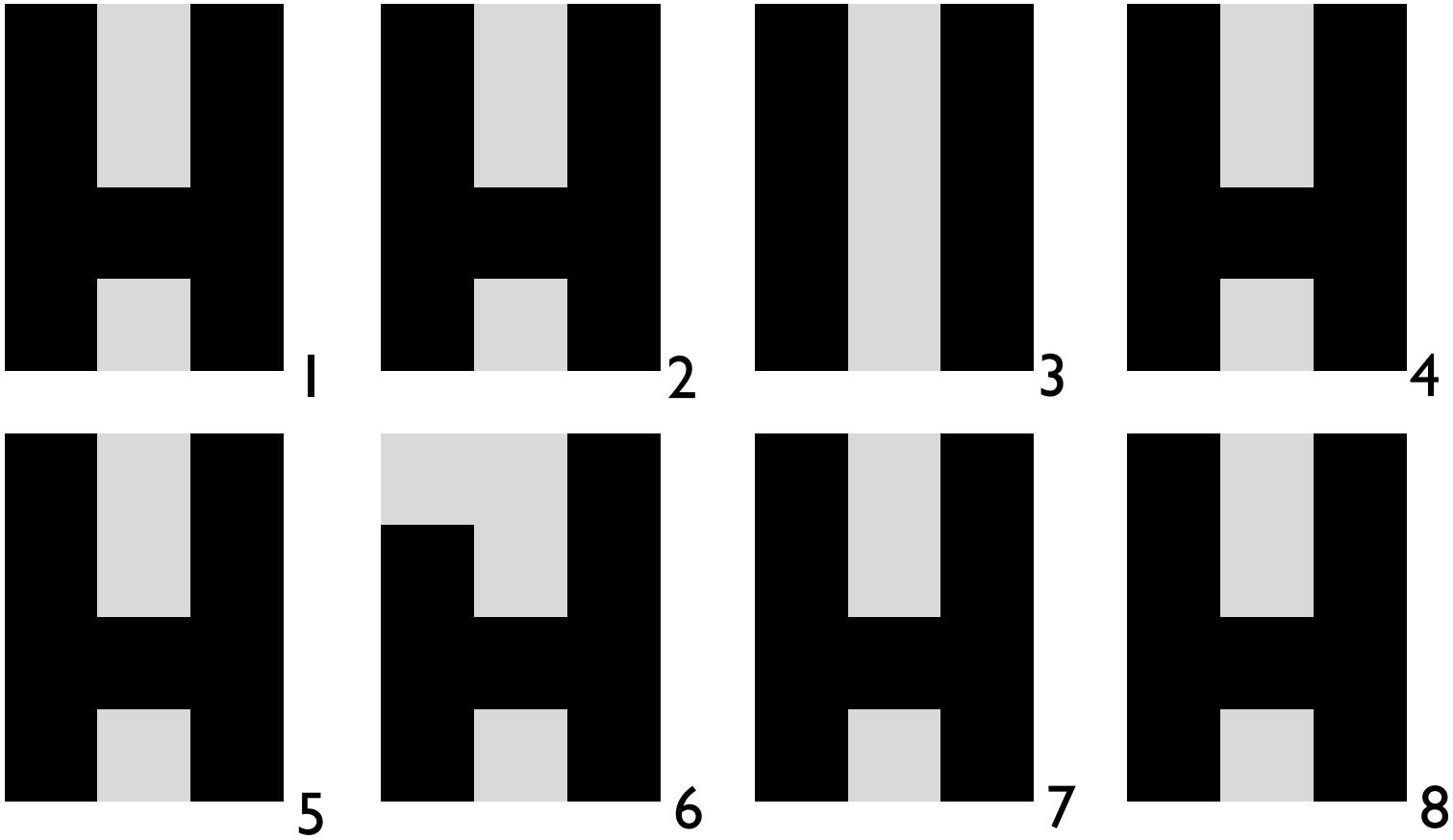
$$P = \begin{bmatrix} 1 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & \mathbf{1} & 1 & 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 & 1 & 1 & \mathbf{0} & 1 & 1 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Avaliação

- Avaliando a aptidão da população resultante ao final dessa geração temos
  - $\boldsymbol{f} = [5, 9, 7, 5, 7, 10, 6, 6]$
  - $f_T = 55$ (aptidão total)
- Aptidão média pode ser calculada por $f_{AV} = f_T / N$
  - N é a quantidade de indivíduos.
  - Geração anterior:
    - $f_{AV} = 47/8 = 5{,}875$
  - Geração atual:
    - $f_{AV} = 55/8 = 6{,}875$
  - Melhoria na aptidão média!
    - Da mesma forma, Distância de Hamming aproxima-se mais de zero.

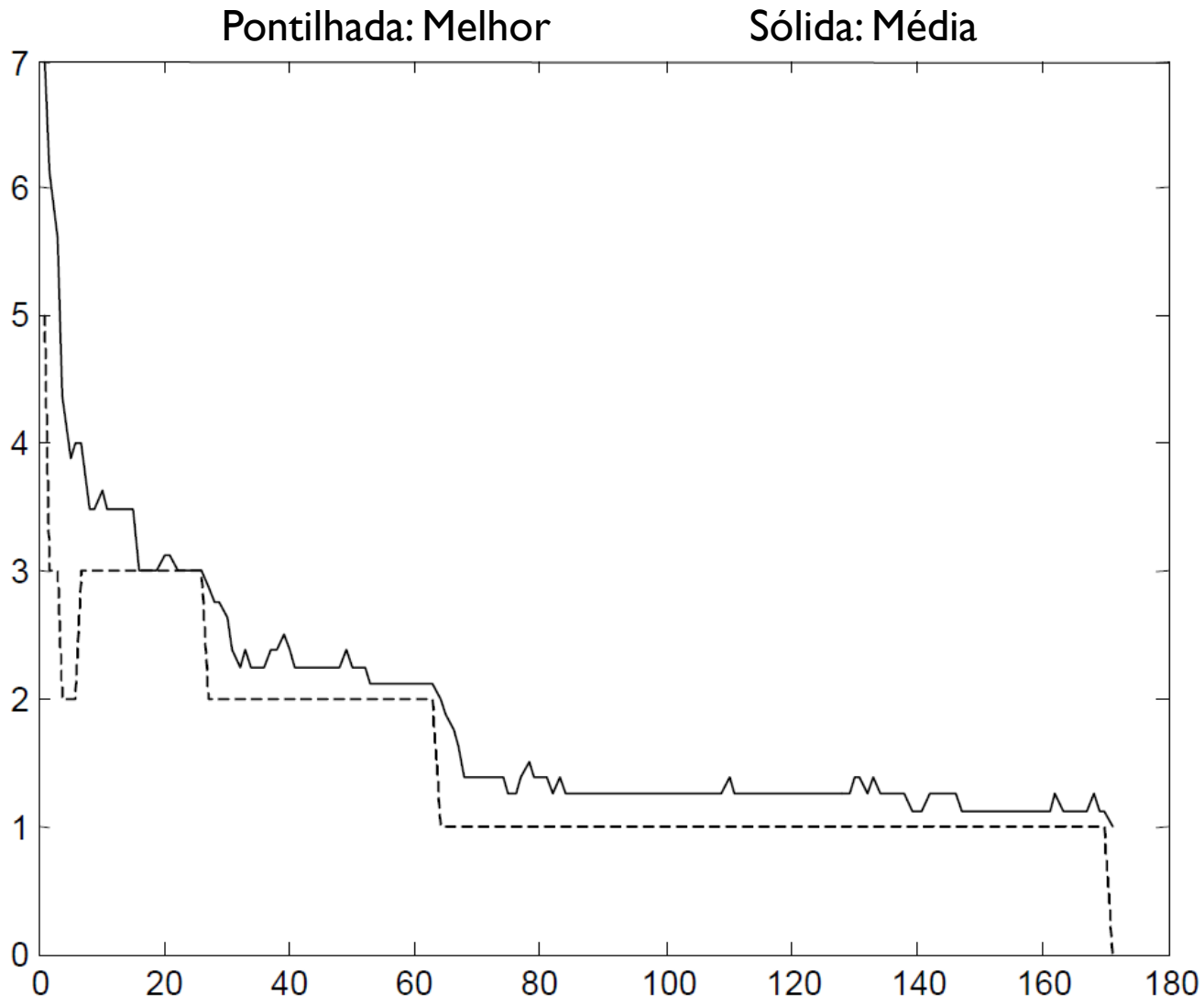
População Final

- População final após 171 gerações:



Evolução da população

- Distância de *Hamming* em cada geração:

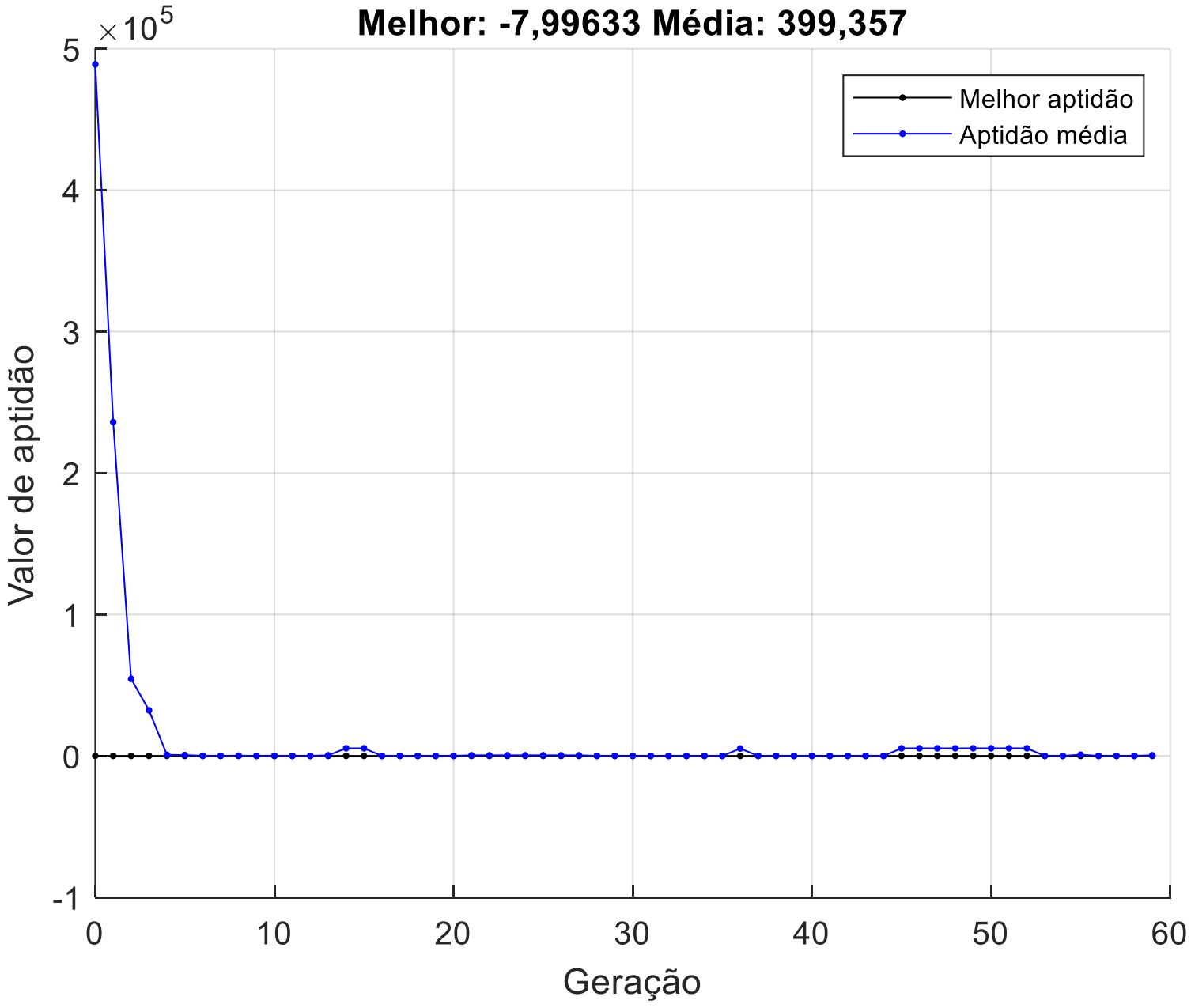


Exemplo 2: Minimização de Polinômio de 4º Grau

- Encontrar valor de x para o qual a função $f(x) = 2x^4 - 3x^3 + 7x - 5$ assume o valor mínimo.
  - Assumir que $x \in (-32, +32)$
  - Codificar X como vetor binário de 16 bits:
    - 1 bit para o sinal (1 - positivo, 0 - negativo).
    - 5 bits para a parte inteira.
    - 10 bits para a parte decimal.
      - Precisão de aproximadamente $0{,}001$ em decimal.

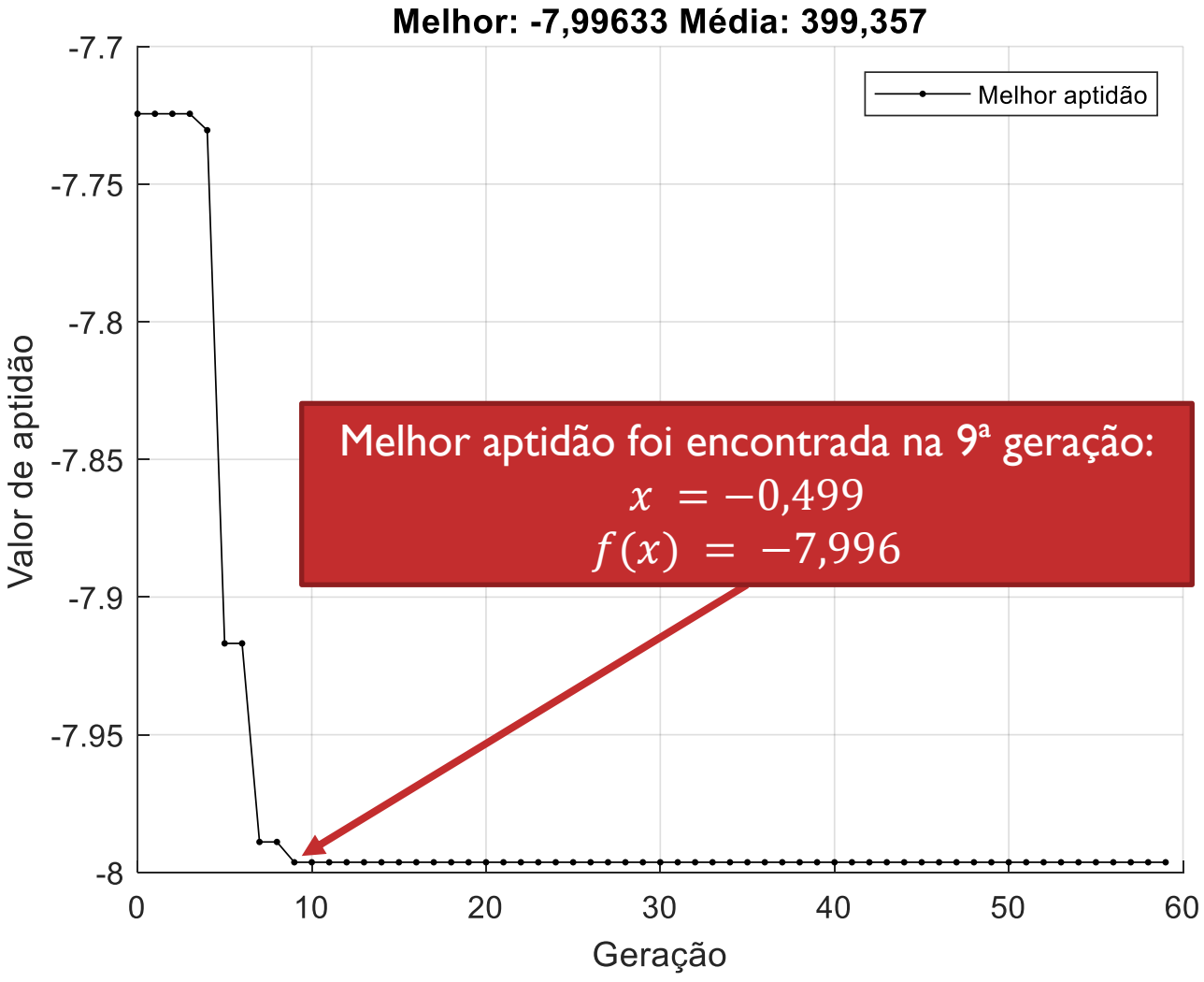
Hiperparâmetros

- População:
  - 30 indivíduos.
- Aptidão:
  - Medida por ranking.
- Seleção:
  - Amostragem universal estocástica.
- Elitismo:
  - Dois melhores indivíduos são mantidos para a próxima geração.

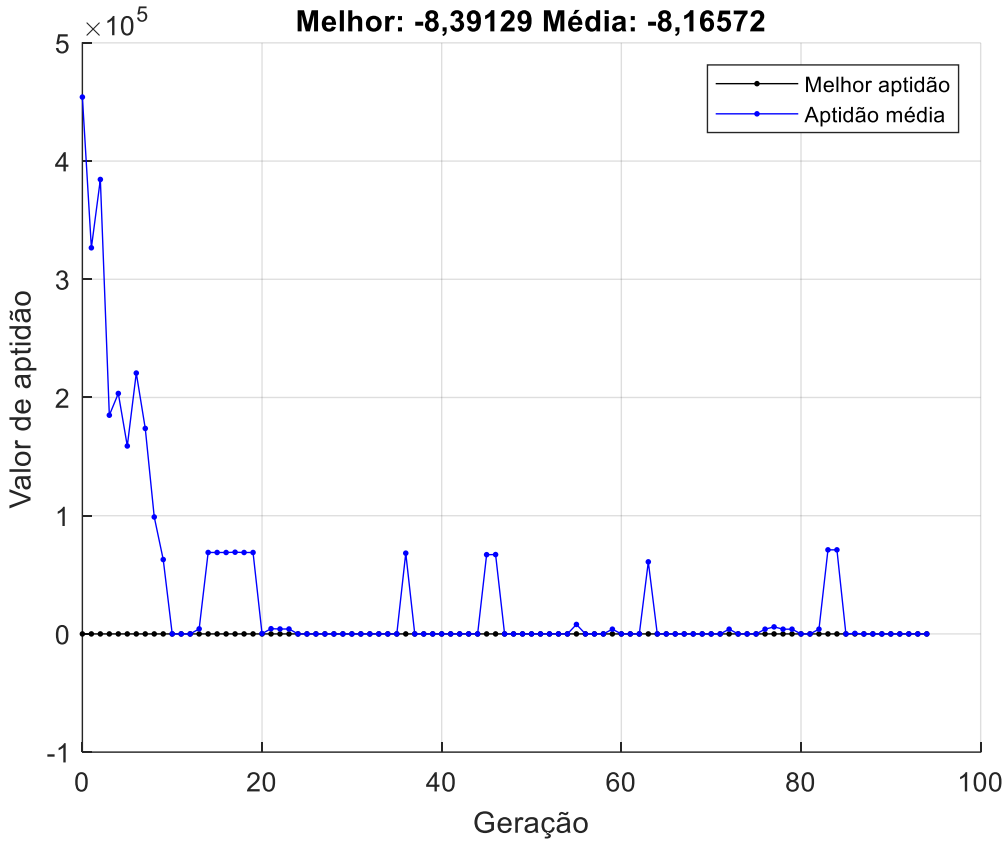

Melhor: -7,99633 Média: 399,357
×10^5
5
4
3
2
1
0
-1
Valor de aptidão
0
10
20
30
40
50
60
Geração
Melhor aptidão
Aptidão média

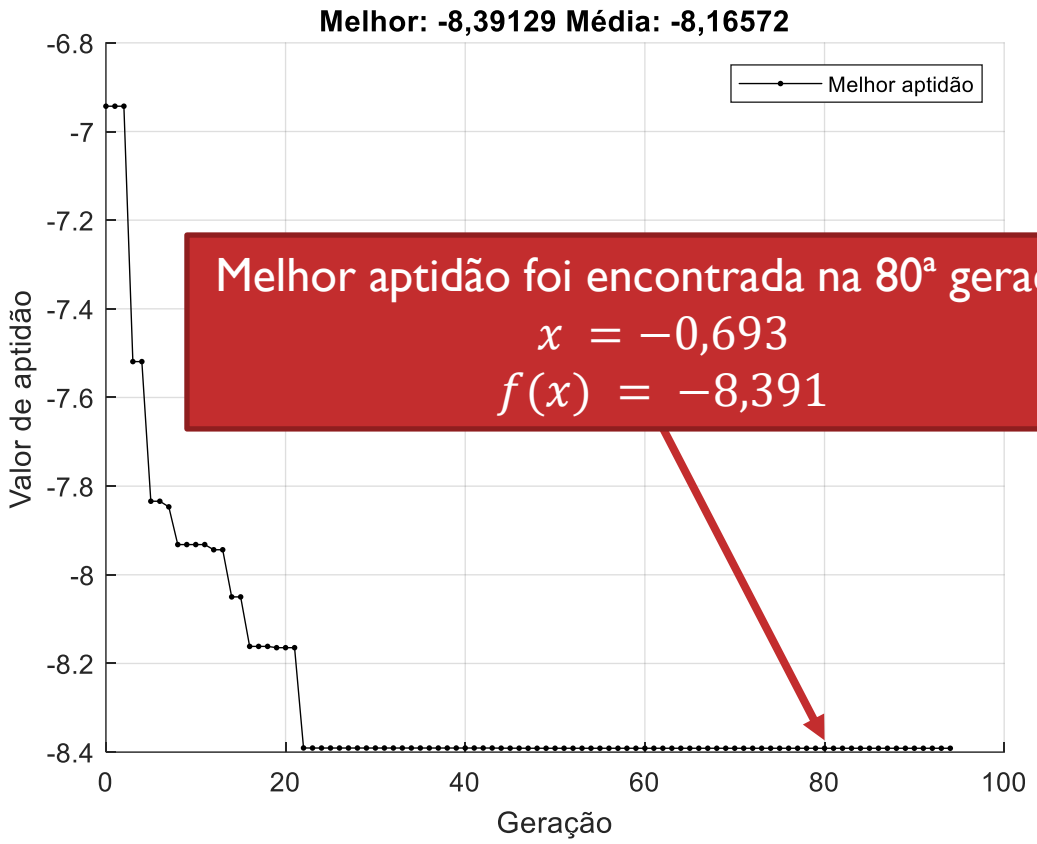
População Final:

0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1
0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1
0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1
0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1
0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1
0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1
0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1
0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1
0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1
0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1
0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1
0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1
0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1
0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1
0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1
0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1
0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1
0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1
0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1
0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1
0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1
0	0	0	0	0	0	0	1	1	1	1	1	0	1	1	1
0	0	1	0	0	0	0	1	1	1	1	1	1	1	1	1
0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1
0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1
0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1
0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1
0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1



Vamos usar o Código de Gray





Melhor aptidão foi encontrada na 80ª geração:

$x = -0{,}693$

$f(x) = -8{,}391$

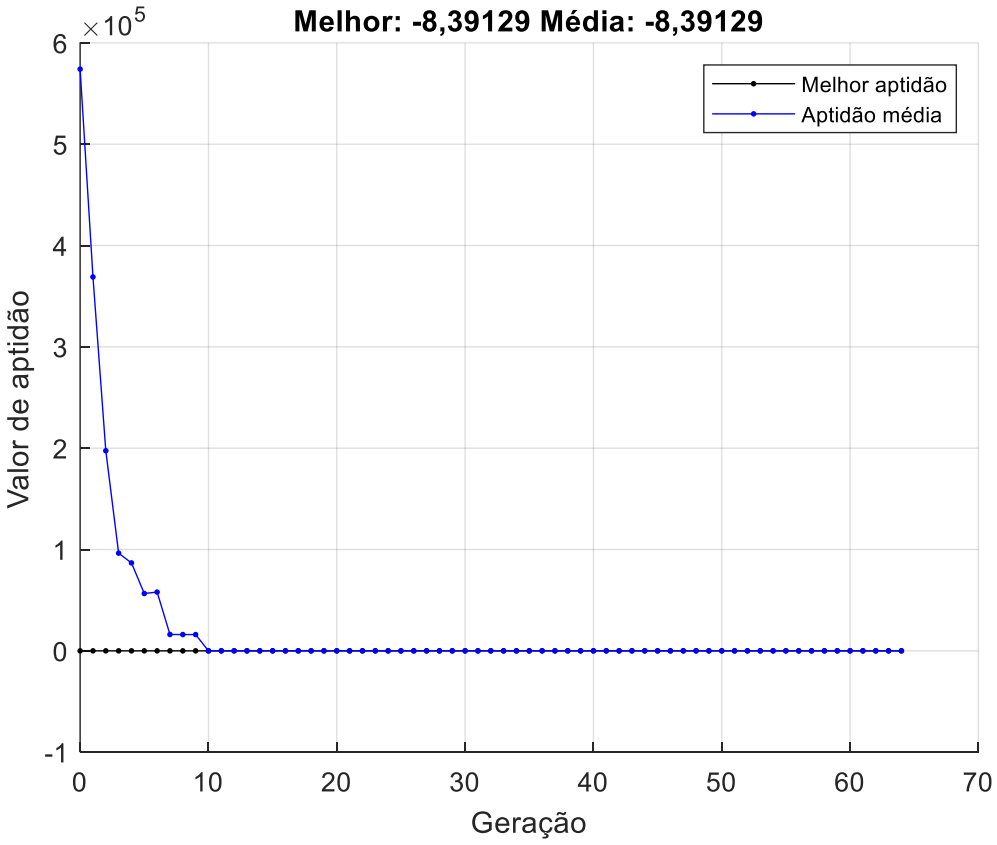
Código de Gray – População Final

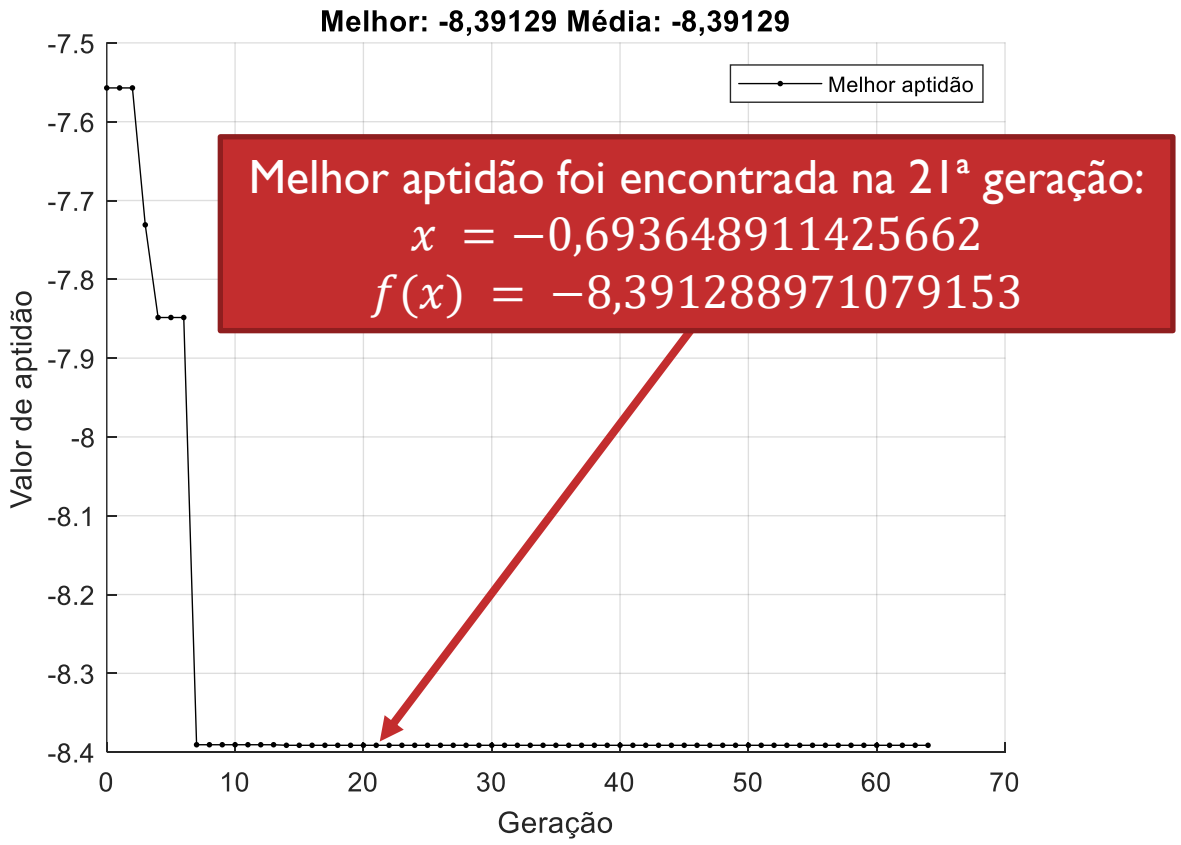
0	0	0	0	0	0	1	1	1	0	1	0	0	1	0	1
0	0	0	0	0	0	1	1	1	0	1	0	0	1	0	1
0	0	0	0	0	0	1	1	1	0	1	0	0	1	0	1
0	0	0	0	0	0	1	1	1	0	1	0	0	1	0	1
0	0	0	0	0	0	1	1	1	0	1	0	0	1	0	1
0	0	0	0	0	0	1	1	1	0	1	0	0	1	0	1
0	0	0	0	0	1	1	1	1	0	1	0	0	1	1	1
0	0	0	0	0	0	1	1	1	0	1	0	0	1	0	1
0	0	0	0	0	0	1	1	1	0	1	0	0	1	0	1
0	0	0	0	0	0	1	1	1	0	1	0	0	1	0	1
0	0	0	0	0	0	1	1	1	0	1	0	0	1	0	1
0	0	0	0	0	0	1	1	1	0	1	0	0	1	0	1
0	0	0	0	0	0	1	1	1	0	1	0	0	1	0	1
0	0	0	0	0	0	1	1	1	0	1	0	0	1	0	1
0	0	0	0	0	0	1	1	1	0	1	0	0	1	0	1
0	0	0	0	0	0	1	1	1	0	1	0	0	1	0	1
0	0	0	0	0	0	1	1	1	0	1	0	0	1	0	1
0	0	0	0	0	0	1	1	1	0	1	0	0	1	0	1
0	0	0	0	0	0	1	1	1	0	1	0	0	1	0	1
0	0	0	0	0	0	1	1	1	0	1	0	0	1	0	1
0	0	0	0	0	0	1	1	1	0	1	0	0	1	0	1
0	0	0	0	0	0	1	1	1	0	1	0	0	1	0	1
0	0	0	0	0	0	1	1	1	0	1	0	0	1	0	1
0	0	0	0	0	0	1	1	1	0	1	0	0	1	0	1
0	0	0	0	0	0	1	1	1	0	1	0	0	1	0	1
0	0	0	0	0	0	1	1	1	0	1	0	0	1	0	1
0	0	0	0	0	0	1	1	1	0	1	0	0	1	0	1
0	0	0	0	0	0	1	1	1	0	1	0	0	1	0	1
0	0	0	0	0	0	1	1	1	0	1	0	0	1	0	1
0	0	0	0	0	0	1	1	1	0	1	0	0	1	0	1
0	0	0	0	0	0	1	1	1	0	1	0	0	1	0	1

Vamos usar uma Representação de Números Reais, com a Restrição $x \in [-32, +32]$

População Final:

-0,693648911425662
-0,693648911425662
-0,693648911425662
-0,693770981738162
-0,693648911425662
-0,693648911425662
-0,693648911425662
-0,693648911425662
-0,693648911425662
-0,693648911425662
-0,693648911425662
-0,693648911425662
-0,693648911425662
-0,693648911425662
-0,693648911425662
-0,693770981738162
-0,693648911425662
-0,693648911425662
-0,693648911425662
-0,693648911425662
-0,693648911425662
-0,693648911425662
-0,693648911425662
-0,693648911425662
-0,693648911425662
-0,693648911425662
-0,693526841113162
-0,693648911425662
-0,693404770800662
-0,693526841113162
-0,693526841113162
-0,693526841113162

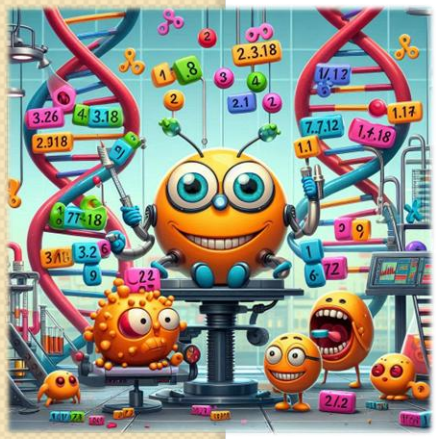




Melhor aptidão foi encontrada na 21ª geração:
$x = -0{,}693648911425662$
$f(x) = -8{,}391288971079153$

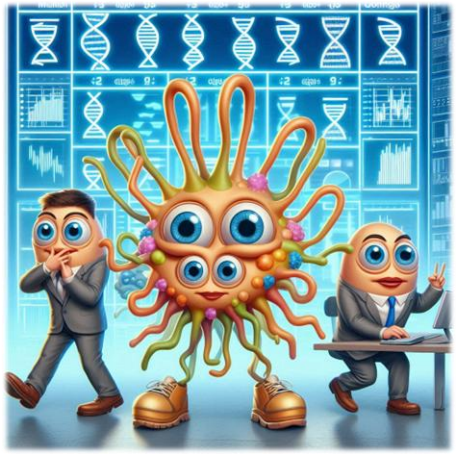
Crossover com Números Reais

- Para cada coordenada (locus) o gene do filho pode ser a média ponderada dos genes dos pais.
  - Os pesos são aleatórios.
  - Exemplo (implementação do MATLAB):
    - $\mathbf{filho} = \mathbf{pai1} + \mathbf{rand} * (\mathbf{pai2} - \mathbf{pai1})$
      - $\mathbf{rand}$ é um vetor de valores aleatórios.
        - Se todos os valores de $\mathbf{rand}$ estiverem no intervalo [0, 1], os filhos são produzidos no hipercubo definido colocando os pais em vértices opostos.
      - $*$ é a multiplicação ponto a ponto dos vetores.
      - Função de *crossover* padrão do MATLAB para problemas com restrições lineares.
        - Há outras funções para diferentes tipos de restrições.



Mutação com Números Reais

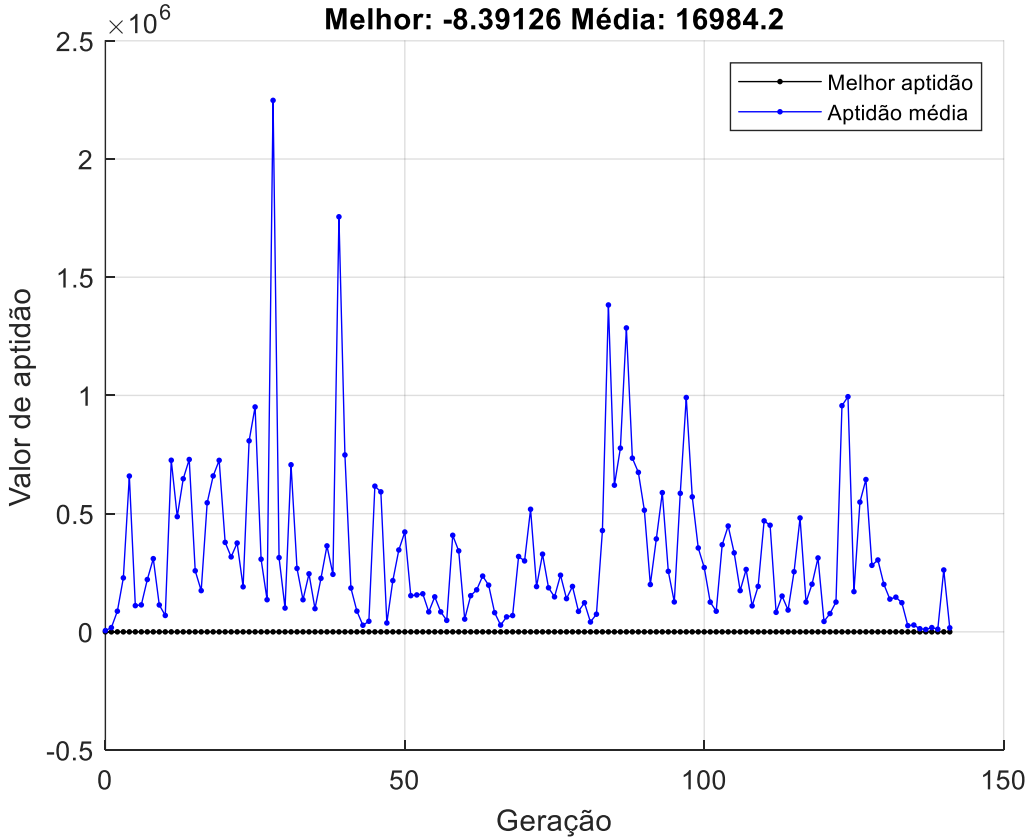
- Pode-se adicionar um vetor aleatório de uma distribuição Gaussiana com média zero ao vetor que representa cada indivíduo.
  - Respeitando fronteiras se existirem.
  - É a função padrão do MATLAB para problemas sem restrições.

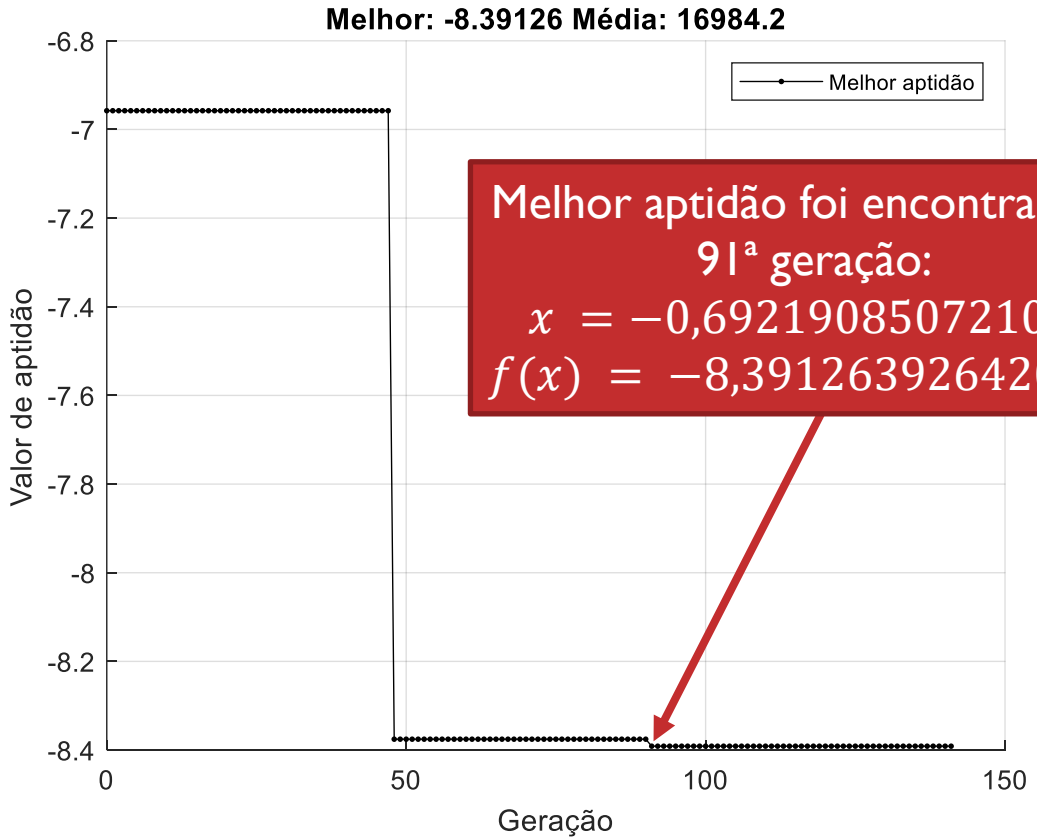


Vamos Remover as Restrições (Fronteiras)

População Final:

-0,692190850721087
-0,692190850721087
-0,692190850721087
-0,692190850721087
-0,692190850721087
-0,692190850721087
-0,692190850721087
-0,692190850721087
-0,692190850721087
-0,692190850721087
-0,692190850721087
-0,692190850721087
-0,692190850721087
-0,692190850721087
-0,692190850721087
15,855803535389800
-0,692190850721087
-0,692190850721087
-0,692190850721087
-0,692190850721087
-0,692190850721087
-0,692190850721087
-0,692190850721087
-0,692190850721087
-0,692190850721087
-10,377906283144800
14,379782690863000
-16,536992791901500
5,195234660997550
1,350955844690010
14,522673660303300
12,835362144364000





Melhor aptidão foi encontrada na 91ª geração:

$x = -0{,}692190850721087$

$f(x) = -8{,}391263926420798$

Média de 200 Execuções

- Lembre-se que estamos tratando de algoritmos estocásticos, então vamos ver como cada variante se sai em uma média de 200 execuções:

Representação	Média da Melhor Aptidão	Melhor das Melhores Aptidões	Pior das Melhores Aptidões
Binária - Tradicional	-8,27602898926061	-8,39128805833752	-5,00000000000000
Binária - Código de Gray	-7,94114862272632	-8,39128805833752	-3,28906250000000
Real com intervalo definido	-8,39128895891030	-8,39128897329092	-8,39128892983324
Real sem restrições	-8,32410003424395	-8,39128889522856	-7,18803211704392

Representação	Média de Gerações	Mínimo de Gerações	Máximo de Gerações
Binária - Tradicional	66,730	51	125
Binária - Código de Gray	84,825	51	208
Real com intervalo definido	62,315	51	73
Real sem restrições	81,445	51	210

Variando o Tamanho da População

- $x \in (-32, +32)$
- 16 bits (1 bit para o sinal, 5 bits para a parte inteira, 10 bits para a parte decimal)

Tamanho da População	Média da Melhor Aptidão	Mediana da Melhor Aptidão	Melhor das Melhores Aptidões	Pior das Melhores Aptidões
10	92,1460705959527	-8,32692447186855	-8,39128805833752	17513,5422668457
30	-8,27602898926061	-8,39068426983249	-8,39128805833752	-5
50	-8,37110250745987	-8,39125513285399	-8,39128805833752	-7

Tamanho da População	Média de Gerações	Mediana de Gerações	Mínimo de Gerações	Máximo de Gerações
10	116,315	108	52	301
30	66,730	60	51	125
50	58,045	56	51	141

Tamanho da População	Média da Melhor Aptidão	Mediana da Melhor Aptidão	Melhor das Melhores Aptidões	Pior das Melhores Aptidões
10	92,1460705959527	-8,32692447186855	-8,39128805833752	17513,5422668457
30	-8,27602898926061	-8,39068426983249	-8,39128805833752	-5
50	-8,37110250745987	-8,39125513285399	-8,39128805833752	-7

Tamanho da População	Média de Gerações	Mediana de Gerações	Mínimo de Gerações	Máximo de Gerações
10	116,315	108	52	301
30	66,730	60	51	125
50	58,045	56	51	141

Tempos de Execução:

Tamanho da População	Tempo de Execução Médio (s)	Tempo de Execução Mediano (s)	Tempo de Execução Mínimo (s)	Tempo de Execução Máximo (s)
10	1,6801795740	1,52792310	0,5986847	4,1226455
30	3,5059251880	3,30845280	2,1401567	7,3779833
50	6,1206916375	6,13871355	4,0491244	15,5157009

Tempos de execução no MATLAB R2023a em um computador desktop com processador AMD Ryzen 9 3900X, 64 GB de RAM e Sistema Operacional Windows 11.

Tamanho da População	Tempo de Execução Médio (s)	Tempo de Execução Mediano (s)	Tempo de Execução Mínimo (s)	Tempo de Execução Máximo (s)
10	0,6163485955	0,58710885	0,2705649	1,5330780
30	0,6118202570	0,55737225	0,4656686	1,1296032
50	0,5785487250	0,56015980	0,5023357	1,3684702

Tempos de execução no MATLAB R2024a em um computador desktop com processador Intel Core i9-14900K, 64 GB de RAM e Sistema Operacional Windows 11.

Variando Tamanho da Elite

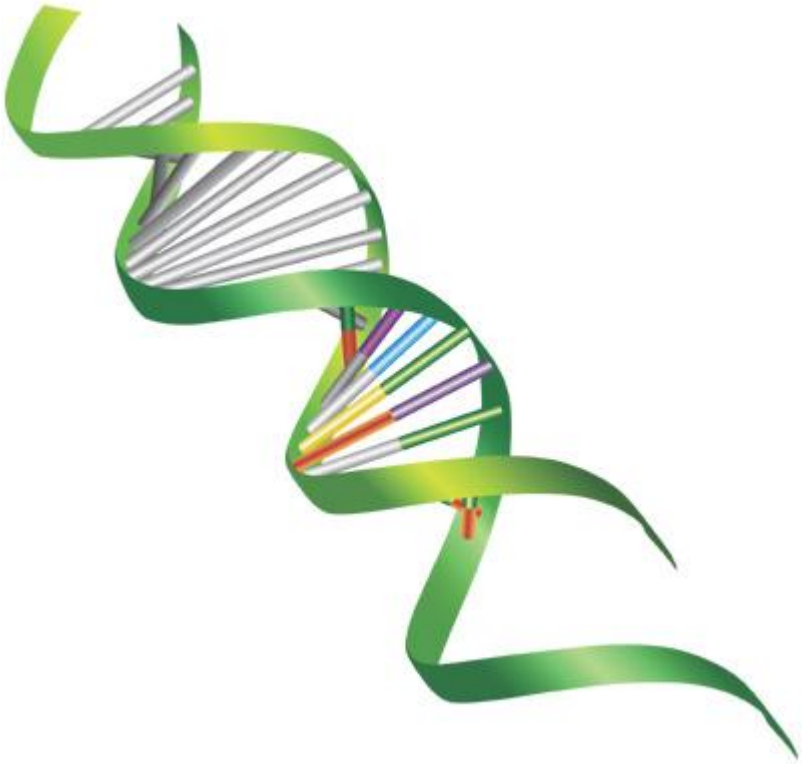
- $x \in (-32, +32)$
- 16 bits (1 bit para o sinal, 5 bits para a parte inteira, 10 bits para a parte decimal)
- População: 30 indivíduos

Tamanho da Elite	Média da Melhor Aptidão	Mediana da Melhor Aptidão	Melhor das Melhores Aptidões	Pior das Melhores Aptidões
0	-8,34174801354291	-8,39125513285399	-8,39128805833752	-5
2	-8,27602898926061	-8,39068426983249	-8,39128805833752	-5
5	-8,26855359104568	-8,39068426983249	-8,39128805833752	-5

Tamanho da Elite	Média de Gerações	Mediana de Gerações	Mínimo de Gerações	Máximo de Gerações
0	65,775	59	51	153
2	66,730	60	51	125
5	69,630	60	51	148

Aplicações

- Otimização de função numérica.
- Otimização combinatória:
  - Problema do caixeiro viajante.
  - Problema de empacotamento.
  - Alocação de recursos.
- Projetos:
  - Projeto de pontes.
- Aprendizado de Máquina:
  - Jogos.



Minhas Aplicações

- Otimizar hiperparâmetros em algoritmos
  - Competição e Cooperação entre Partículas e seus derivados.
- Encontrar os melhores componentes (atributos de pixels) para segmentar imagens.
  - BREVE, Fabricio Aparecido. **Interactive Image Segmentation using Label Propagation through Complex Networks**. *Expert Systems With Applications*, v. 123, p.18 – 33, 2019.
    - https://doi.org/10.1016/j.eswa.2019.01.031

Table 2
Preliminary study on a larger feature set with 9 images from the GrabCut dataset and weights optimized using a Genetic Algorithm.

Image / Feature	dog	21077	124084	271008	208001	llama	doll	person7	sheep	teddy	Mean	
Row	0.99	0.96	0.99	0.44	0.98	0.23	0.97	0.79	0.94	0.44	0.77	(±0.29)
Col	0.80	0.72	0.67	0.91	0.97	1.00	0.34	0.99	0.83	0.39	0.76	(±0.24)
R	0.63	0.03	0.06	0.91	0.45	0.45	0.56	0.58	0.88	0.28	0.48	(±0.30)
G	0.35	0.49	0.25	0.96	0.30	0.28	0.31	0.10	0.54	0.67	0.42	(±0.25)
B	0.96	0.45	0.18	0.36	0.54	0.46	0.26	0.19	0.80	0.39	0.46	(±0.25)
H	0.03	0.04	0.08	0.37	0.07	0.74	0.03	0.15	0.11	0.14	0.17	(±0.22)
S	0.74	0.10	0.06	0.09	0.10	0.30	0.21	0.07	0.10	0.92	0.27	(±0.31)
V	0.65	0.28	0.71	0.94	0.80	0.24	0.18	0.61	0.74	0.29	0.54	(±0.27)
ExR	0.96	0.21	0.06	0.85	0.84	0.79	0.13	0.38	0.12	0.47	0.48	(±0.35)
ExB	0.72	0.36	0.64	0.70	0.42	0.47	0.16	0.83	0.35	0.98	0.56	(±0.25)
ExG	0.75	0.10	0.07	0.22	0.19	0.04	0.78	0.29	0.24	0.32	0.30	(±0.26)
MR	0.11	0.38	0.03	0.35	0.13	0.21	0.78	0.34	0.81	0.22	0.34	(±0.27)
MG	0.13	0.23	0.79	0.30	0.76	0.47	0.70	0.18	0.61	0.93	0.51	(±0.29)
MB	0.49	0.31	0.33	0.42	0.45	0.24	0.15	0.29	0.77	0.80	0.42	(±0.21)
SDR	0.01	0.12	0.06	0.11	0.08	0.38	0.02	0.27	0.20	0.22	0.15	(±0.12)
SDG	0.01	0.08	0.09	0.03	0.09	0.38	0.21	0.02	0.27	0.06	0.12	(±0.12)
SDB	0.00	0.05	0.05	0.04	0.22	0.22	0.06	0.40	0.13	0.01	0.12	(±0.13)
MH	0.58	0.04	0.16	0.91	0.15	0.92	0.03	0.27	0.14	0.86	0.40	(±0.37)
MS	0.65	0.31	0.04	0.06	0.21	0.21	0.36	0.89	0.41	0.67	0.38	(±0.28)
MV	0.02	0.95	0.04	0.35	0.55	0.78	0.66	0.57	0.80	0.25	0.50	(±0.32)
SDH	0.17	0.41	0.19	0.39	0.48	0.32	0.07	0.31	0.38	0.08	0.28	(±0.14)
SDS	0.03	0.41	0.22	0.15	0.18	0.23	0.13	0.50	0.51	0.02	0.24	(±0.18)
SDV	0.61	0.21	0.07	0.10	0.13	0.31	0.03	0.10	0.38	0.24	0.22	(±0.18)

Exercício

- Encontrar valor de x para o qual a função $f(x) = x^2 - 4x + 4$ assume o valor mínimo.
  - Assumir que $x \in [-15, +15]$
  - Codificar X como vetor binário.
    - Usar 5 bits, primeiro bit é o sinal (1 - positivo, 0 - negativo).
  - Criar uma população inicial com 4 indivíduos.
  - Utilizando *crossover* de um ponto e mutação em apenas um gene da população.
  - Definir o valor mínimo após no máximo 10 gerações.

Esta Foto de Autor Desconhecido está licenciado em CC BY-ND

Subida da Colina, Recozimento Simulado e Algoritmos Genéticos

- Subida da Colina e Recozimento Simulado:
  - Único indivíduo.
  - Gera novo indivíduo na vizinhança do ponto atual.
- Algoritmos Genéticos:
  - Busca baseada em populações.
  - Podem gerar novos indivíduos na vizinhança de várias possíveis soluções.
  - Processo de competição.
  - Indivíduos com novas características e características dos pais.

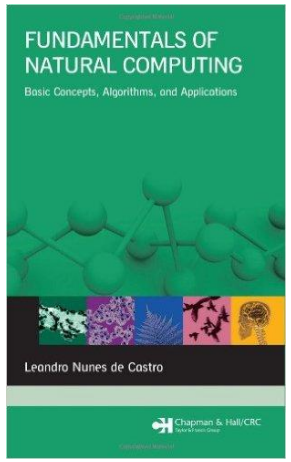
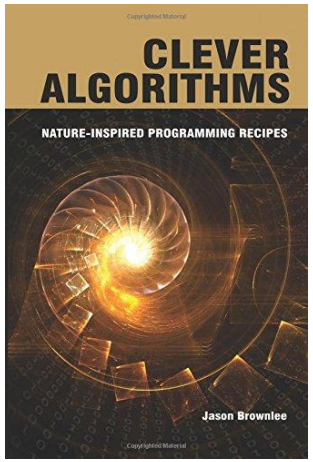
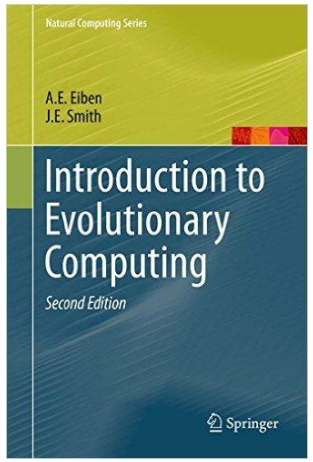
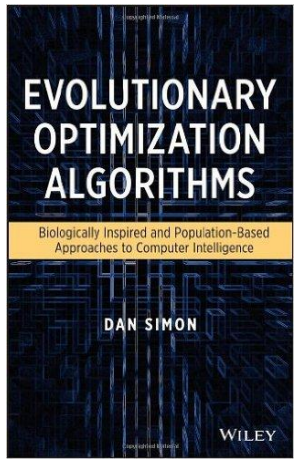
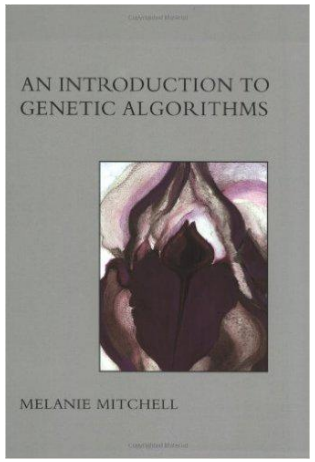
Analogia com Cangurus

- Subida da Colina:
  - Os cangurus podem pular até chegar ao topo da montanha próxima de onde começaram, mas não há garantias de que essa montanha seja o Everest, nem mesmo que seja uma montanha alta.
- Recozimento Simulado:
  - O canguru está bêbado e pula aleatoriamente por um longo período. Aos poucos ele vai ficando sóbrio e tende a pular montanha acima.
- Algoritmos Genéticos:
  - Muitos cangurus são jogados de paraquedas no Himalaia em pontos aleatórios. Estes cangurus não sabem que devem ir até o topo do Everest. Porém, de anos em anos você atira nos cangurus em altitudes baixas e espera que os que restarem se reproduzam e se multipliquem.

Cangurus em Lone Pine Koala Sanctuary – Brisbane, Austrália – por Fabricio Breve

Próxima Aula: Sistema Nervoso

- Introdução à neurocomputação e aos desafios do cérebro.
- Comparação entre capacidades do cérebro e computadores (Deep Blue vs. Kasparov).
- Fundamentos dos neurônios, sinapses e neurotransmissores.
- Estrutura do sistema nervoso (central e periférico).
- Desenvolvimento neural a partir do tubo neural.
- Neuroplasticidade e neurogênese.
- Regiões e funções do cérebro (lobos e hemisférios).
- Transmissão sináptica e sinais elétrico-químicos.
- Organização em camadas e mapas sensoriais.
- Mitos e debates em neurociência.

Bibliografia

- CASTRO, Leandro Nunes. *Fundamentals of Natural Computing: Basic Concepts, Algorithms, And Applications*. CRC Press, 2006.
- CARVALHO, André Ponce de Leon F. de. *Notas de Aula*, 2007.
- BROWNLEE, Jason. *Clever Algorithms: Nature-Inspired Programming Recipes*. Jason Brownlee, 2011.
- EIBEN, A. E.; SMITH, James E. *Introduction to Evolutionary Computing*, 2nd Edition. Springer, 2015.
- SIMON, Dan. *Evolutionary Optimization Algorithms*. Wiley, 2013.
- MITCHELL, Melaine. *An Introduction to Genetic Algorithms*. MIT Press, 1998.