



# Inteligência Artificial

## EC016

### Algoritmos Genéticos

**Prof. Felipe Andery Reis**

**[fandery@inatel.br](mailto:fandery@inatel.br)**

*Material adaptado do prof. Edielson*

**mar/2018**

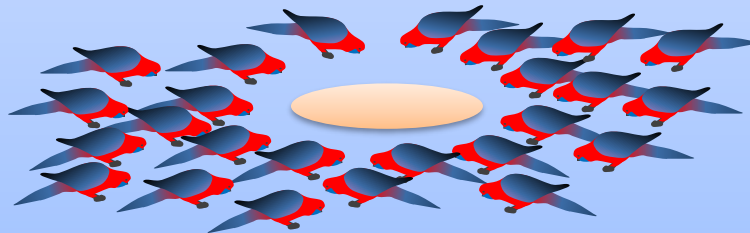
# Algoritmos genéticos

## ➤ Paradigma do raciocínio da IA

### **Evolucionista**

#### **Metáfora da natureza**

ex. algoritmos genéticos, vida artificial,...



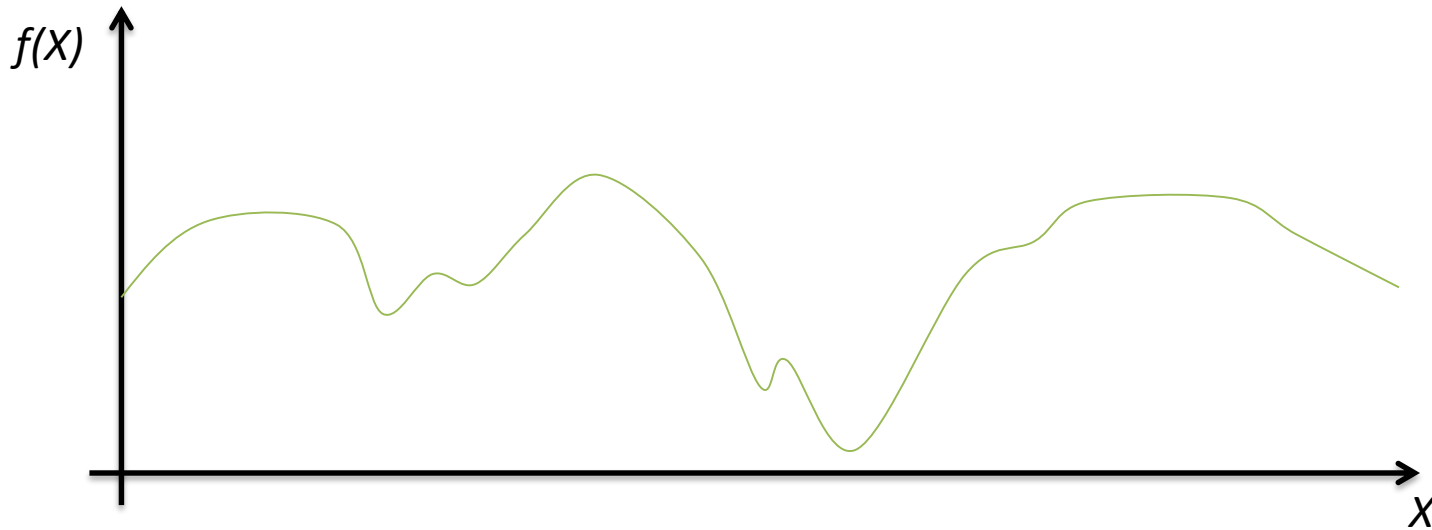
# Algoritmos genéticos

## ➤ Mínimo Local

### ➤ Problema:

Um sistema tem o seu consumo energético dependente do ponto da curva  $f(X)$  abaixo, porém a curva não é conhecida, apenas a função:

```
Void Consumo(float X, float *Esquerda, float *Valor, float *Direita);
```

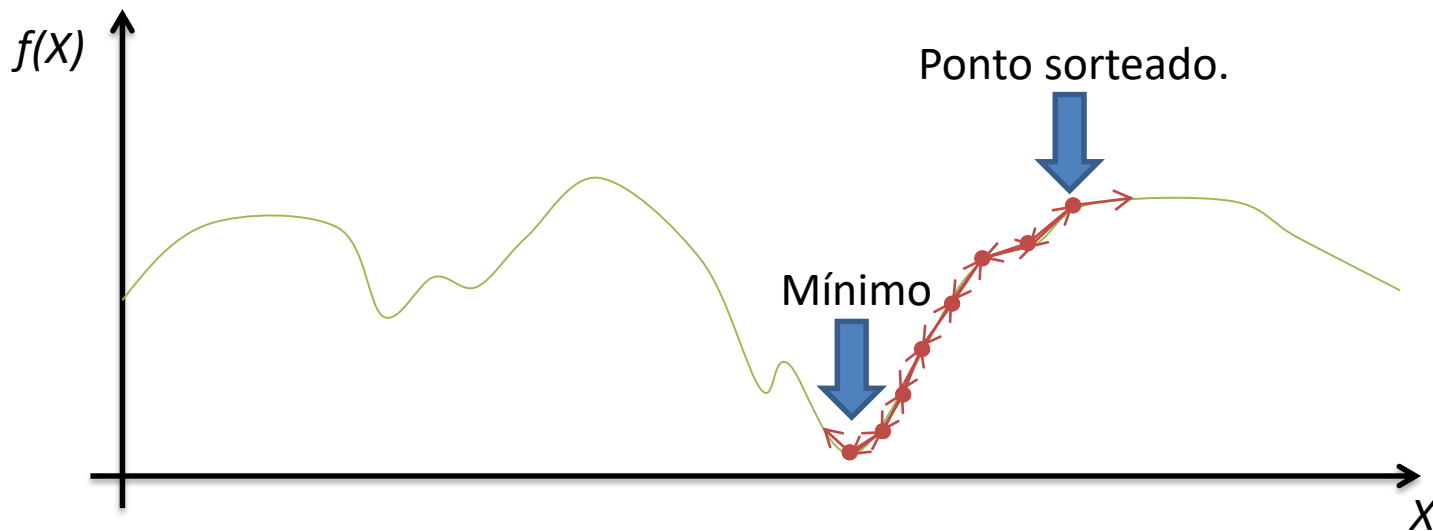


# Algoritmos genéticos

## ➤ Mínimo Local

### ➤ Solução:

1. Parta de um ponto qualquer;
2. Visite os seus vizinhos;
3. Se pelo menos um dos vizinhos for menos que ele, mova para o que tiver menor valor e repita os passos 2 e 3;

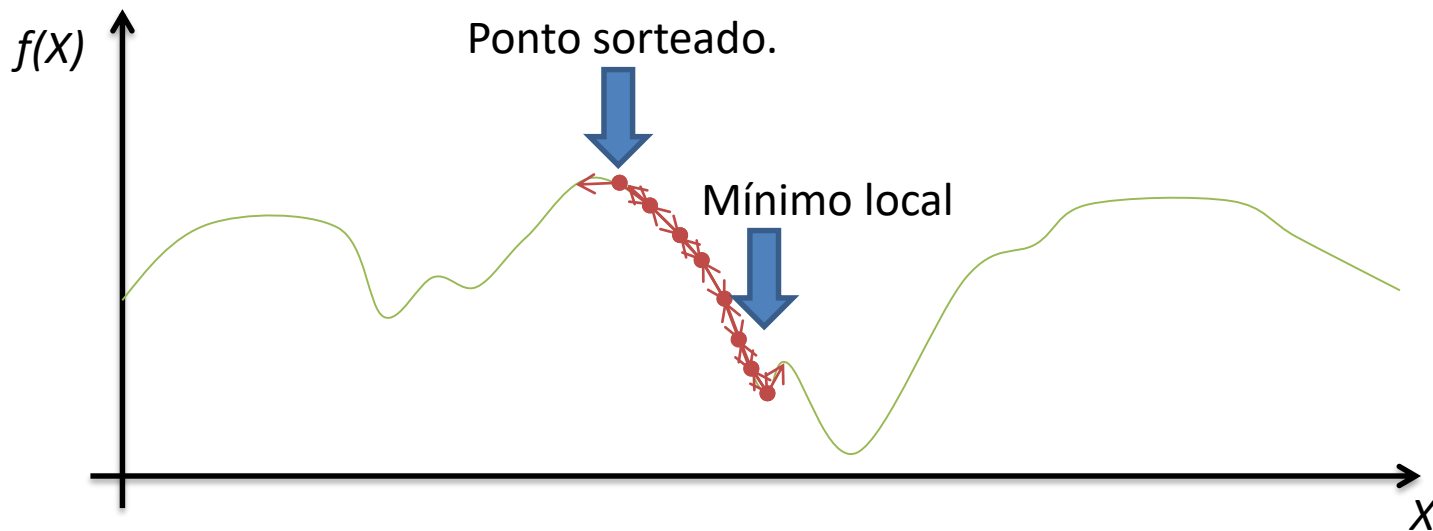


# Algoritmos genéticos

## ➤ Mínimo Local

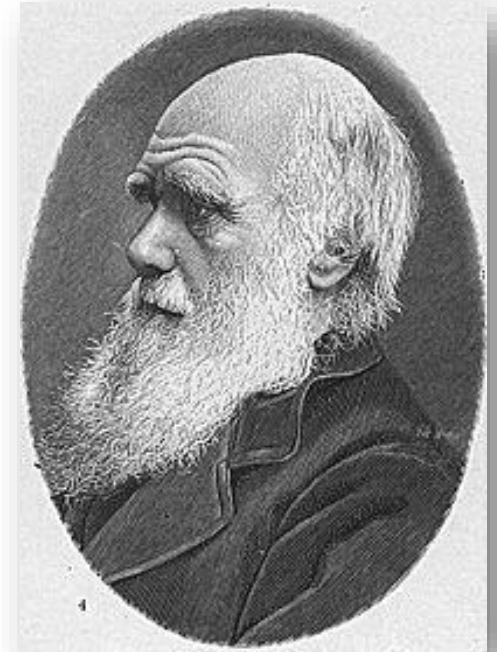
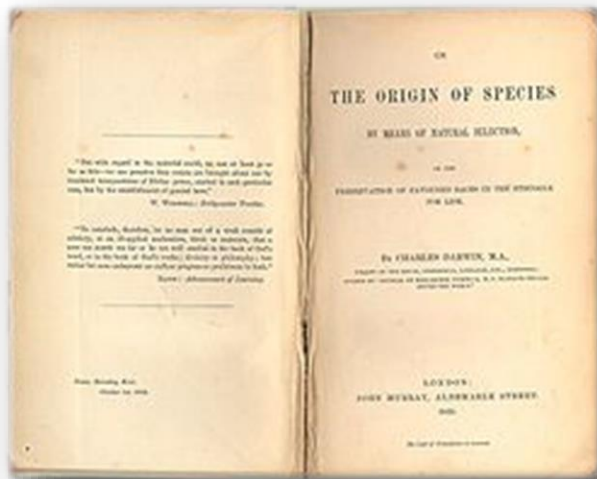
### ➤ Solução:

1. Parta de um ponto qualquer;
2. Visite os seus vizinhos;
3. Se pelo menos um dos vizinhos for menos que ele, mova para o que tiver menor valor e repita os passos 2 e 3;



# Algoritmos genéticos

- Em 1859, 2 anos após George Boole ter descrito sua álgebra, o naturalista Britânico Charles Robert Darwin, publicou sua obra *“On the Origin of Species by Means of Natural Selection”*.



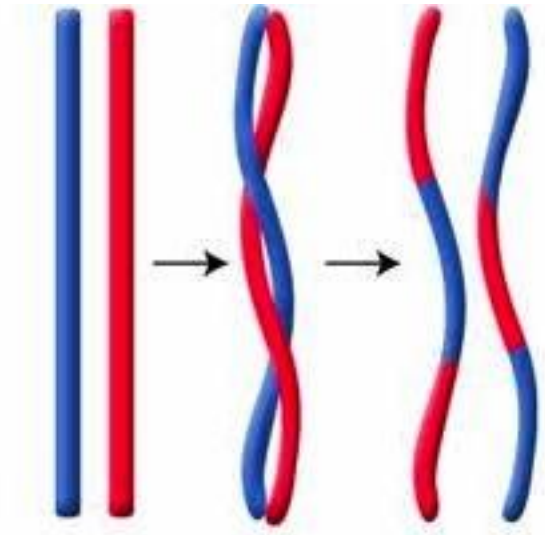
- Origem da genética;
- A natureza incentiva os mais fortes em detrimento dos mais fracos.

# Algoritmos genéticos

## ➤ Algoritmos evolucionários:

Usam modelos computacionais do processos naturais de evolução como uma ferramenta para resolver problemas.

- ✓ Estratégias Evolucionárias
- ✓ Algoritmos Genéticos
- ✓ Programação Genética



# Algoritmos genéticos

## ➤ O que são algoritmos genéticos (GA)?

- Ramo dos algoritmos evolucionários
  - técnica de busca baseada na metáfora do processo biológico de evolução natural.
- Podem ser consideradas técnicas heurísticas de otimização global
  - se opõe a outros métodos que ficam facilmente retidos em mínimos (ou máximos) locais (*hill climbing*).
- Populações de indivíduos são criadas e submetidas aos operadores genéticos:
  - Seleção;
  - Recombinação (*crossover*) e;
  - Mutação.



# Algoritmos genéticos

## ➤ Principais características

- **Probabilística:** mesmas configurações (população inicial e parâmetros) levam a diferentes soluções;
- **Diminuição do espaço de busca:** não avaliam todas as soluções possíveis;
- **Codificação de parâmetros:** generalização do algoritmo para vários problemas.
  - Alteração apenas da função de avaliação;
  - Economia de tempo e dinheiro.

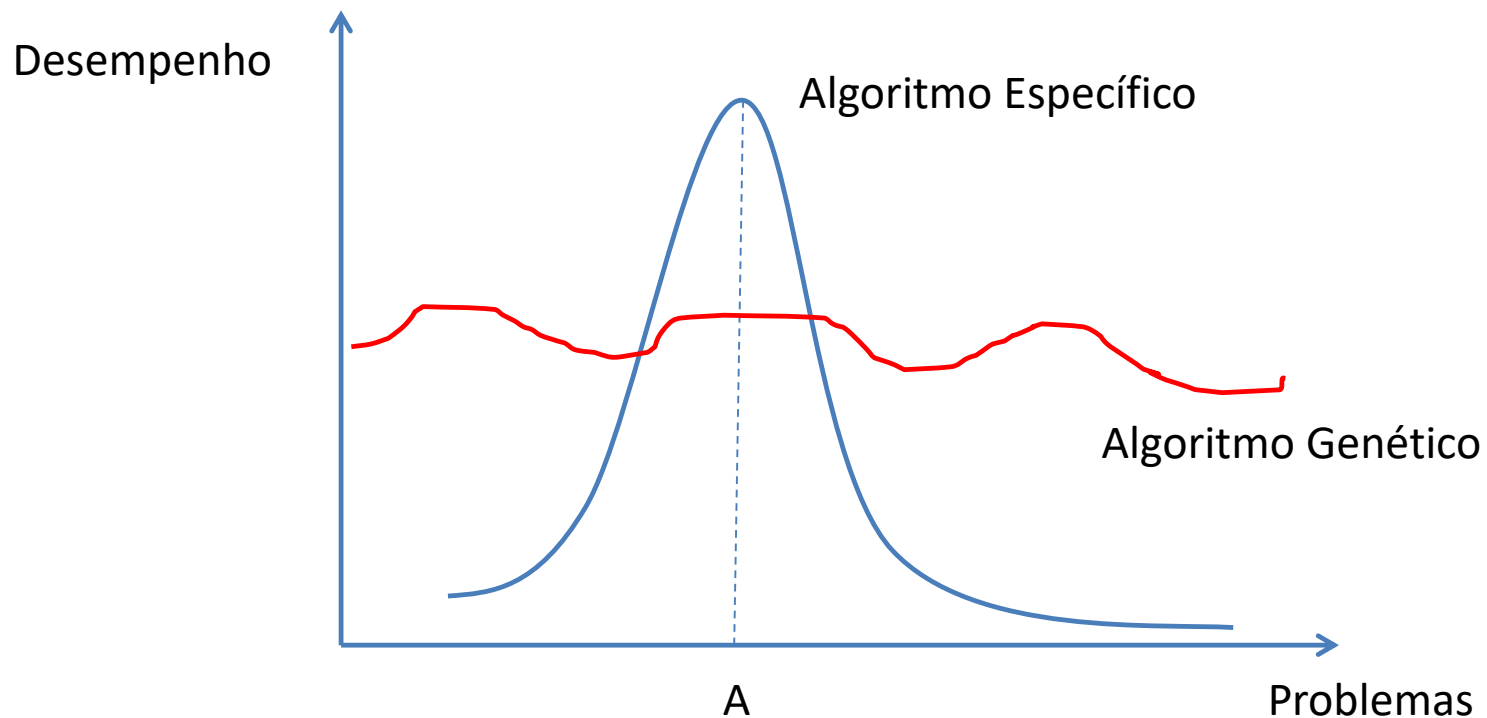
# Algoritmos genéticos

## ➤ Por que GAs?

- **Paralela:** mantém uma população de soluções que são avaliadas simultaneamente;
- **Global:** não usa apenas informação local (menor chance de ficar preso em mínimos locais);
- **Não totalmente aleatória:** possui componentes aleatórios mas os próximos passos são baseados em informações da população corrente;
- **Não afetada por descontinuidades:** não utilizam informações de derivadas na sua evolução;
- Atua em funções discretas e contínuas;

# Algoritmos genéticos

## ➤ Algoritmo Específico x GA

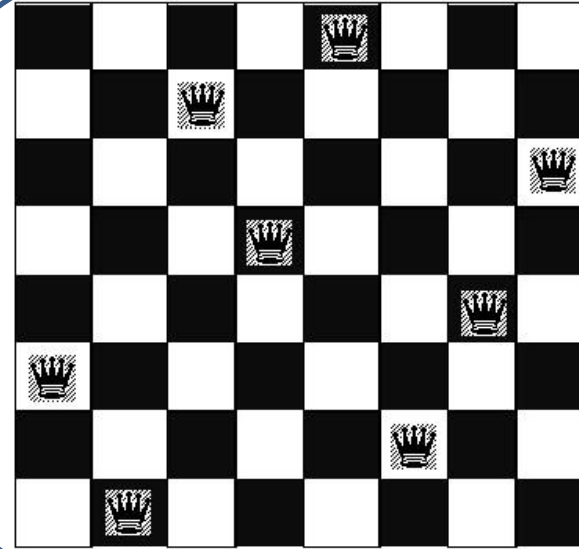


# Algoritmo Genético

# Algoritmos genéticos

## ➤ Terminologia

fenótipo



cromossomo/indivíduo

3 1 7 5 8 2 4 6

genótipo

gene

alelo: possíveis valores {1,2,3,4,5,6,7,8}

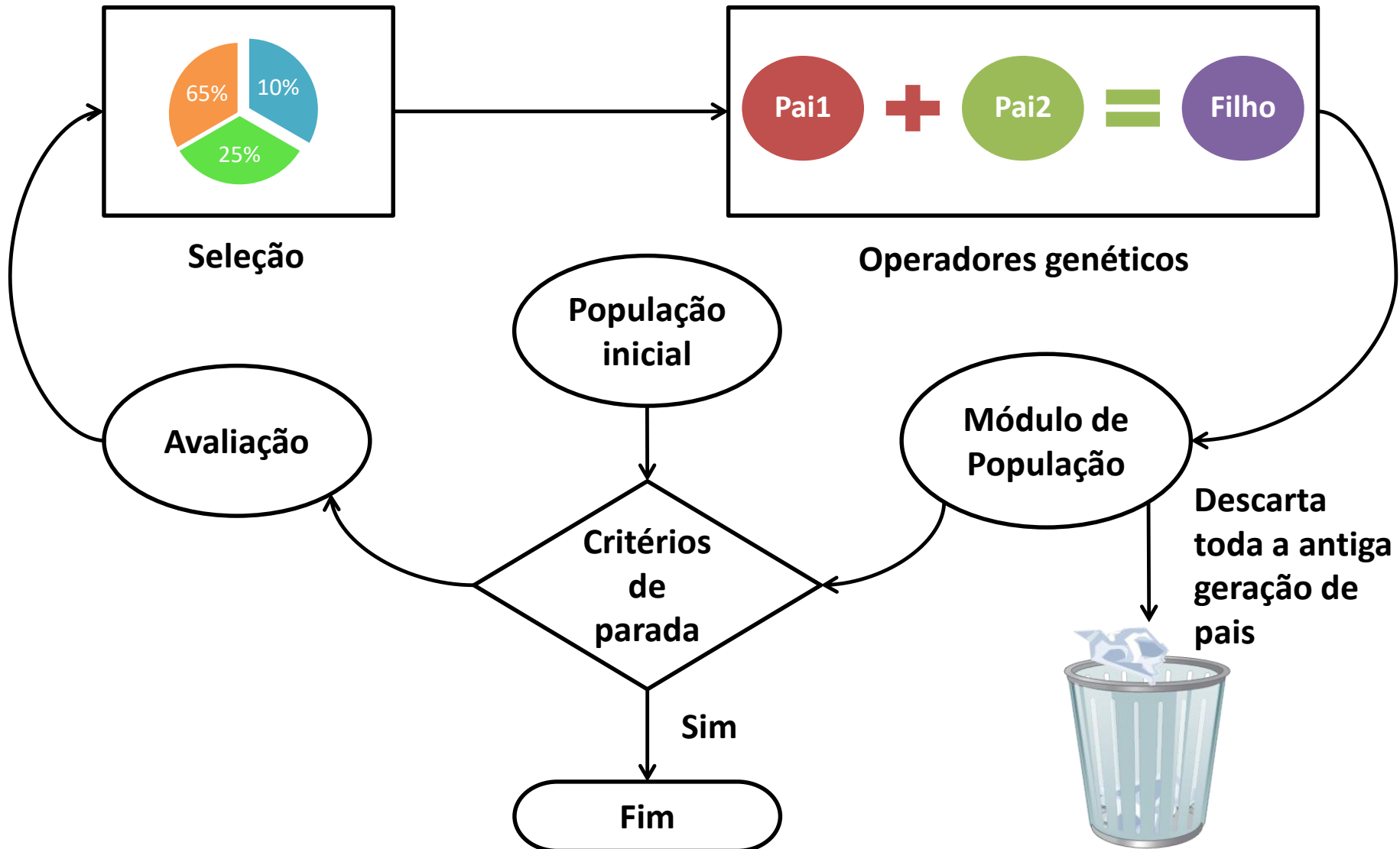
locus: possíveis posições {0,1,2,3,4,5,6,7}

# Algoritmos genéticos

## ➤ Passo a passo de um algoritmo genético:

1. Inicialize a população de cromossomos;
2. Avalie cada cromossomo na população;
3. Selecione os pais para gerar novos cromossomos;
4. Aplique os operadores de recombinação e mutação a estes pais de forma a gerar os indivíduos da nova geração;
5. Apague os velhos membros da população;
6. Avalie todos os novos cromossomos e insira-os na população;
7. Se o tempo acabou, ou o melhor cromossomo satisfaz os requerimentos e desempenho, retorne-o, caso contrário, volte ao passo 3.

# Algoritmos genéticos



# Algoritmos genéticos

## ➤ Componentes do diagrama

- **Seleção:** escolhe-se os indivíduos que participarão do processo de reprodução;
- **Operadores genéticos:** aplica-se os operadores de recombinação e mutação aos indivíduos escolhidos para “pais”;
- **Módulo de população:** define-se a nova população a partir da geração existente e dos filhos gerados;
- **Critério de parada:** verifica-se se o critério de parada é satisfeito. Pode ser por número de gerações ou qualidade da solução;
- **Avaliação:** aplica-se a função de avaliação a cada um dos indivíduos desta geração;



# Algoritmos genéticos

## ➤ Algoritmo

**function** GENETIC-ALGORITHM(population, FITNESS-FN) **returns** an individual

**inputs:** *population*, a set of individuals

FITNESS-FN, a function that measures the fitness of an individual

**repeat**

    new population  $\leftarrow$  empty set

**for**  $i = 1$  **to** SIZE(population) **do**

$x \leftarrow$  RANDOM-SELECTION(population, FITNESS-FN)

$y \leftarrow$  RANDOM-SELECTION(population, FITNESS-FN)

        child  $\leftarrow$  REPRODUCE( $x$ ,  $y$ )

**if** (small random probability) **then** child  $\leftarrow$  MUTATE(child)

        add child to new population

    population  $\leftarrow$  new population

**until** some individual is fit enough, or enough time has elapsed

**return** the best individual in population, according to FITNESS-FN

**function** REPRODUCE( $x$ ,  $y$ ) **returns** an individual

**inputs:**  $x$ ,  $y$ , parent individuals

$n \leftarrow$  LENGTH( $x$ );  $c \leftarrow$  random number from 1 to  $n$

**return** APPEND(SUBSTRING( $x$ , 1,  $c$ ), SUBSTRING( $y$ ,  $c + 1$ ,  $n$ ))

# **Algoritmo Genético**

## **Representação Cromossomial**

# Algoritmos genéticos

## ➤ Representação cromossomial

- Cada pedaço indivisível da representação é chamado de **gene**
  - Deve ser a mais simples possível
  - Soluções proibidas não deve possuir representação
  - Imposições devem estar implícitas dentro da representação

Genótipo	Fenótipo	Problema
0010101001110101	10869	Otimização numérica
CGDEHABF	comece pela cidade C, depois passe pelas cidades G, D, E, H, A, B e termine em F	Caixeiro viajante
$C_1R_4C_2R_6C_4R_1$	se condição 1 ( $C_1$ ) execute regra 4 ( $C_4$ ), se ( $C_2$ ) execute ( $R_6$ ), se ( $C_4$ ) execute ( $R_1$ )	Regras de aprendizagem para agente

# Algoritmos genéticos

## ➤ Representação binária

- Precisão de uma representação binária de uma variável  $x_i$ , que existe no intervalo  $[inf_i, sup_i]$

$$Precisão = \frac{sup_i - inf_i}{2^k - 1}$$

- Conversão do número binário dentro do cromossomo para um número real

$$real = inf_i + \frac{sup_i - inf_i}{2^k - 1} r_i$$

Onde  $r_i$  representa o número inteiro correspondente ao valor binário dentro do cromossomo.

# Algoritmos genéticos

## ➤ Exemplo:

- Seja o problema de encontrar o máximo da função abaixo, sendo que ambas as variáveis  $x$  e  $y$  pertencem ao intervalo dado por  $[-100, 100]$

$$f(x, y) = |x * y * \text{sen}(y\pi/4)|$$

## ▪ Solução:

- Representação binária de 44 bits, sendo 22 bits para cada variável;
- Faixa de valores decimais:  $[0, \dots, 2^{22} - 1]$
- $\text{Precisão} = \frac{\text{sup}_i - \text{inf}_i}{2^k - 1} = \frac{100 - (-100)}{2^{22} - 1} = 4,768e^{-5}$
- $\text{Real} = \text{inf}_i + \frac{\text{sup}_i - \text{inf}_i}{2^k - 1} r_i = 4,768e^{-5} r_i - 100$

# Algoritmos genéticos

## ➤ Outras representações:

- Representação em ordem

- casos onde a ordem é importante para o problema (por exemplo: grafo, caixeiro viajante)
- normalmente todos os nós devem estar presentes em um cromossomo

2 4 7 4 8 5 5 2

- Representação numérica

- evitar a representação binária
- cromossomos que representam diretamente os números reais

2.1 4.5 7.3 4.9 8.1 5.3

# Algoritmo Genético

## Função objetivo

# Algoritmos genéticos

## ➤ Características dos indivíduos:

- **Grau de adaptação:** representa o quão bem a resposta representada por indivíduo soluciona o problema proposto. Calculada pela função objetivo ( $f_o(x)$ ). É uma nota dada ao indivíduo na resolução do problema.
- **Grau de aptidão:** diz respeito ao nível de adaptação de um indivíduo em relação à população à qual ele pertence. Para uma população com  $n$  indivíduos:

$$f_A(x) = \frac{f_o(x)}{\sum_{i=1}^n f_o(i)}$$



# Algoritmo Genético

## Seleção

# Algoritmos genéticos

## ➤ Função de avaliação

- O processo de seleção começa pela função de avaliação
- Determina a qualidade de um indivíduo/cromossomo
  - Deve embutir todo o conhecimento sobre o problema (restrições, qualidades, etc.)
  - Devem refletir os objetivos a serem alcançados
- Exemplo: maximizar a função quadrática  $f(x) = x^2$ , no intervalo de  $x = [0,15]$

Cromossomo	Avaliação/Grau de Adaptação
0001	1
0011	9
0100	16
0110	36
Total	62

# Algoritmos genéticos

## ➤ Seleção

- Deve simular o mecanismo de seleção natural
  - Pais mais capazes geram mais filhos
  - Pais menos aptos também podem gerar descendentes (menor probabilidade), para evitar a **convergência genética**
- Convergência genética
  - Se só os melhores estiverem presentes, somente eles irão se reproduzir. Desta forma a população ficará com indivíduos cada vez mais iguais, causando assim a falta de diversidade para a população.

# Algoritmos genéticos

## ➤ Seleção

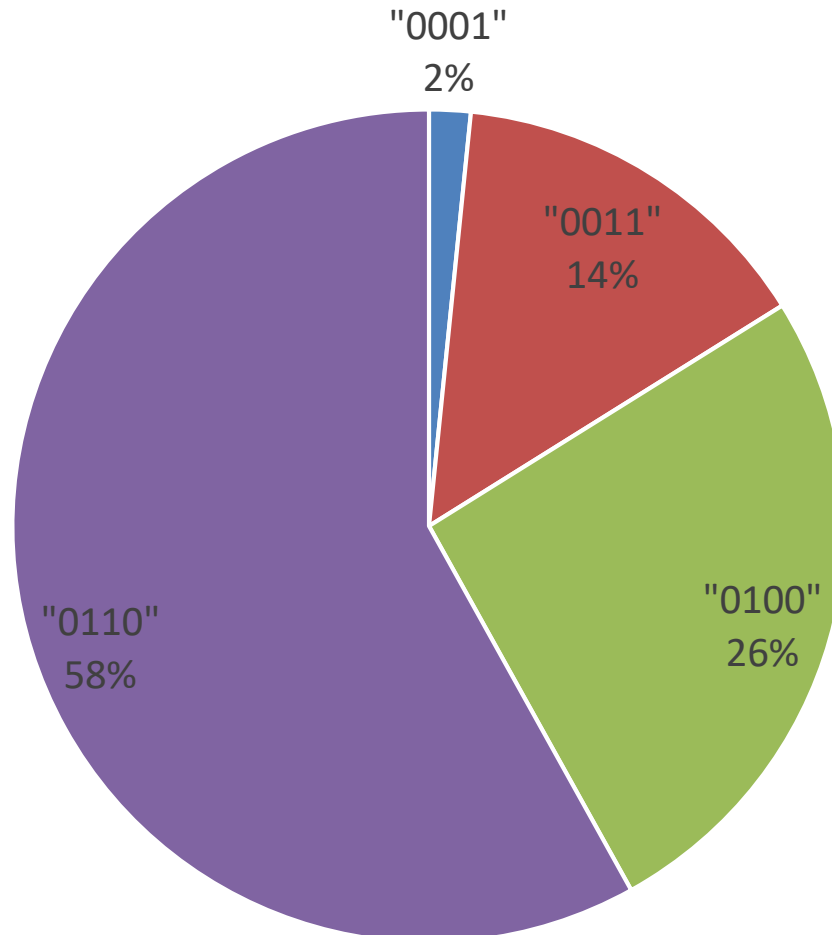
- Método da roleta viciada
  - Cada cromossomo recebe sua chance de ser selecionado
  - Exemplo: maximizar a função quadrática  $f(x) = x^2$ , no intervalo de  $x = [0,15]$

Cromossomo	Adaptação	Grau de Aptidão (%)
0001	1	1,61
0011	9	14,51
0100	16	25,81
0110	36	58,07
Total	62	100,00

# Algoritmos genéticos

## ➤ Seleção

Roleta viciada para a população do exemplo



# Algoritmos genéticos

## ➤ Outros métodos de seleção:

- **Seleção por ranking (*rank selection*)**: os indivíduos da população são ordenados de acordo com seu valor de adequação e então sua probabilidade de escolha é atribuída conforme a posição que ocupam;
- **Seleção por torneio (*tournament selection*)**: grupos de soluções são escolhidos sucessivamente e as mais adaptadas dentro de cada um destes são selecionadas;
- **Seleção uniforme**: todos indivíduos possuem a mesma probabilidade de serem selecionados. Obviamente, esta forma de seleção possui uma probabilidade muito remota de causar uma melhora da população sobre a qual atua.

# Algoritmo Genético

## Operadores genéticos

# Algoritmos genéticos

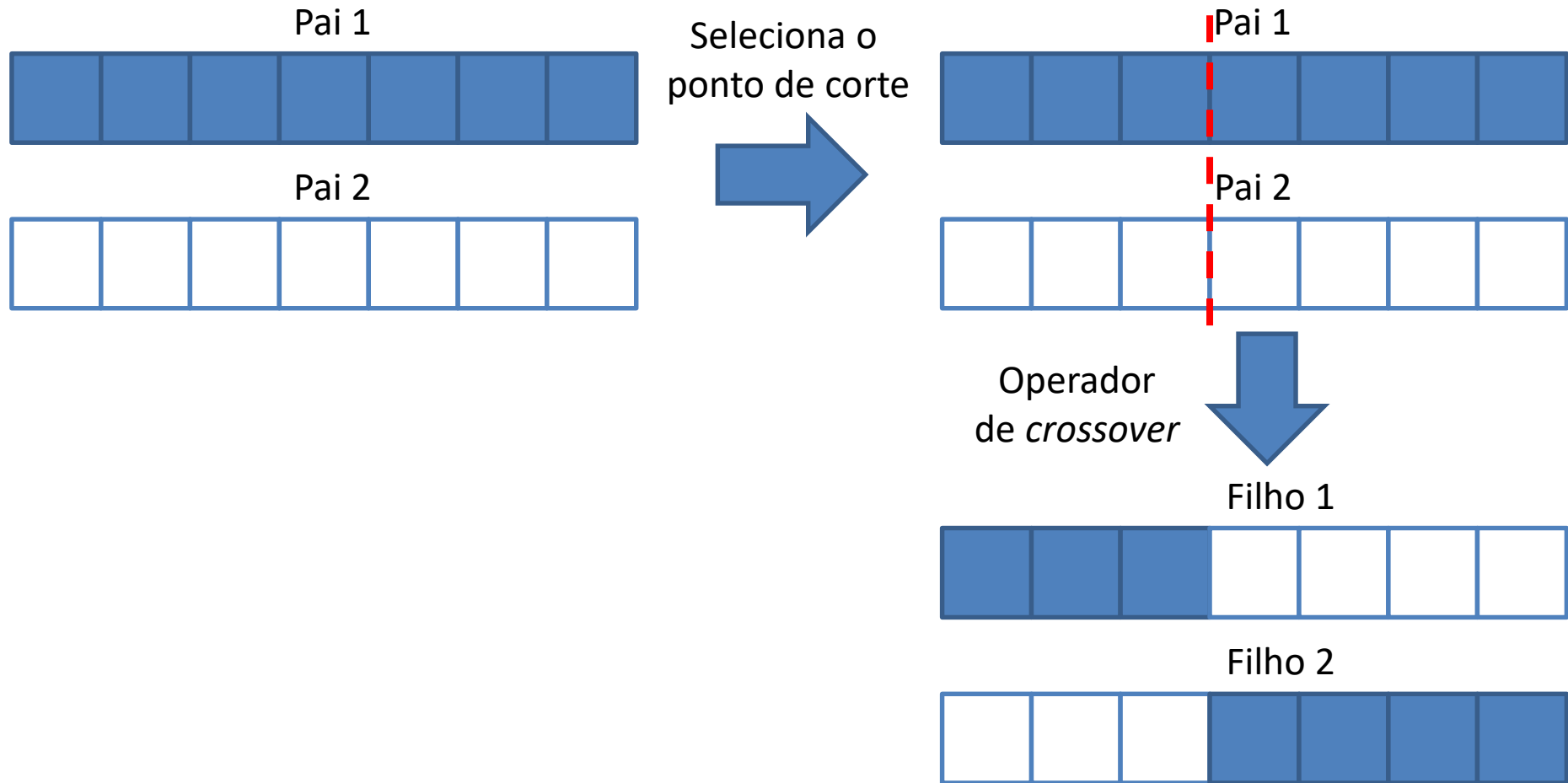
## ➤ Operador de *crossover*

- Responsável por combinar características dos pais selecionados
  - Ponto de corte representa uma posição entre dois genes (cromossomo com  $n$  genes possui  $n - 1$  pontos de corte).
- Método de cruzamento de um ponto
  - Sorteia-se um ponto de corte;
  - O primeiro filho será composto pela concatenação da parte esquerda ao ponto de corte do primeiro pai, e da parte direita ao ponto de corte do segundo pai;
  - O segundo filho será composto pela concatenação da parte direita ao ponto de corte do primeiro pai, e da parte esquerda ao ponto de corte do segundo pai.



# Algoritmos genéticos

## ➤ Exemplo: operador de *crossover*



# Algoritmos genéticos

## ➤ Outros métodos de *crossover*

- **Cruzamento multiponto:** o cruzamento multi-ponto é uma generalização do operador de um ponto. Nele são sorteados um número fixo  $n$  de pontos de corte.
- **Cruzamento segmentado:** o cruzamento segmentado funciona de maneira semelhante ao multi-ponto, com a exceção de que sorteia o número de pontos de corte toda vez que é executado;
- **Cruzamento uniforme:** para cada gene a ser preenchido nos cromossomos filhos, o operador de cruzamento uniforme sorteia de qual dos pais este deve ser gerado.

# Algoritmos genéticos

## ➤ Operador de mutação

- Responsável por alterar características dos indivíduos gerados durante o *crossover*
  - Garante que diversas alternativas serão exploradas, mantendo assim um nível mínimo de abrangência na busca
  - Testa a probabilidade de mutação para todos os genes (probabilidade muito baixa  $\sim 0,5\%$ ). Caso o teste seja positivo, o valor do gene é alterado aleatoriamente.
- Operadores de mutação
  - **mutação *flip***: cada gene a ser mutado recebe um valor sorteado do alfabeto válido;
  - **mutação por troca (*swap mutation*)**: são sorteados  $n$  pares de genes, e os elementos do par trocam de valor entre si;
  - **mutação *creep***: um valor aleatório é somado ou subtraído do valor do gene.

# Algoritmos genéticos

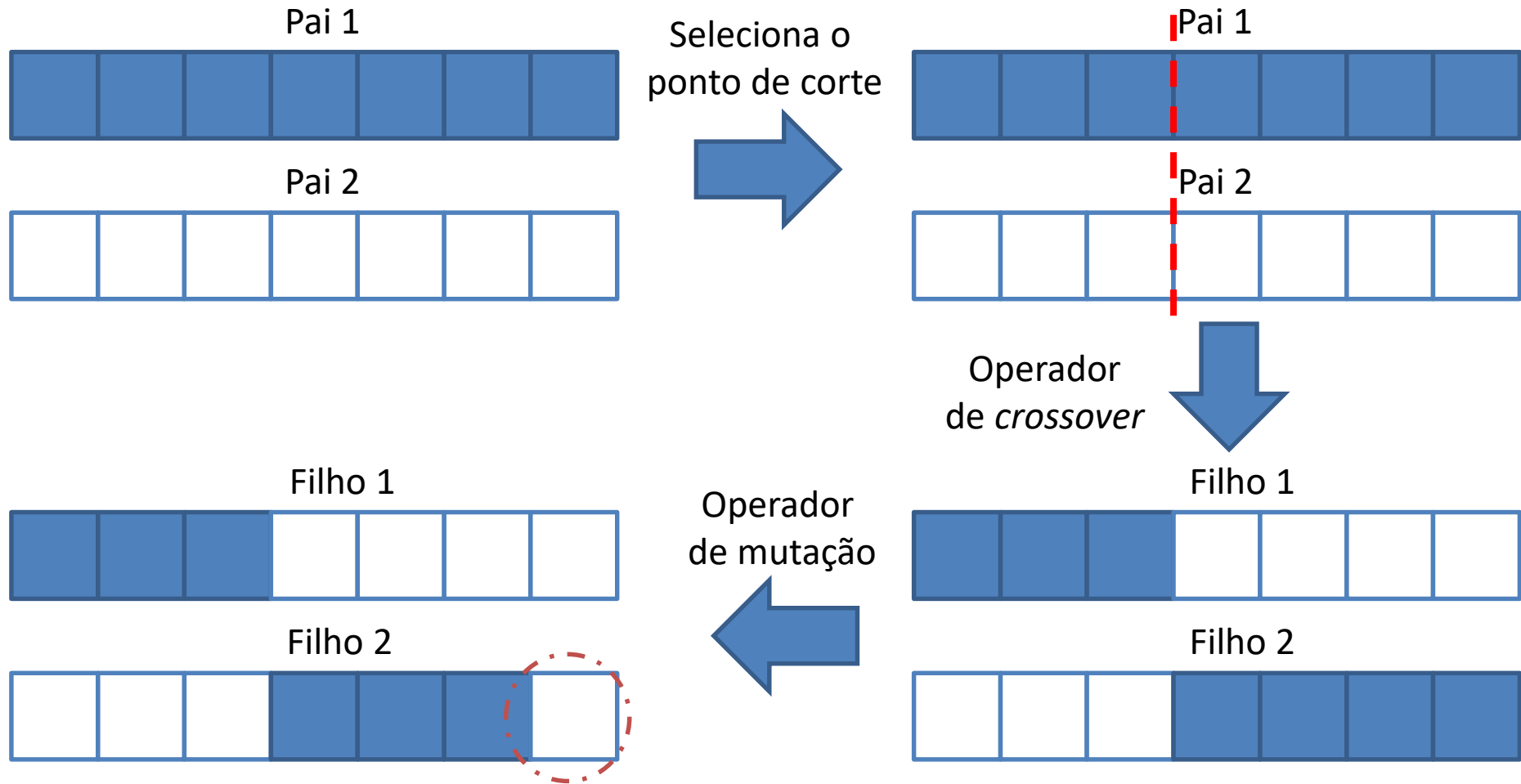
## ➤ Operador de mutação

- Probabilidade de mutação
  - Se probabilidade for muita alta, fará com que o algoritmo genético possua soluções que são determinadas de forma aleatória, ou seja, sem utilizar informações passadas.
- Operador de mutação
  - O operador de mutação é necessário para introdução e manutenção da diversidade genética da população
  - Assegura que a probabilidade de se chegar a qualquer ponto do espaço de busca nunca será zero

# Algoritmos genéticos

## ➤ Exemplo: operador de mutação

### ■ Exemplo:



# Algoritmo Genético

## Módulo de população

# Algoritmos genéticos

## ➤ Módulo de população

- Responsável por controlar a população de indivíduos
  - controla o número de indivíduos, descartando os indivíduos da geração anterior
- Evolução baseada na dinâmica populacional
  - propaga as características desejáveis às gerações subsequentes (cruzamento)
  - manipula a frequência com que determinadas sequências de genes aparecem nas populações

# Algoritmos genéticos

## ➤ População: características

- **Geração:** número de vezes que a população passou pelo processo de seleção, reprodução, mutação e atualização
- **Média de adaptação:**
  - $$M_A = \frac{\sum_{i=1}^n f_O(i)}{n}$$
- **Grau de convergência:** mede a proximidade da média de adaptação da atual geração em relação às anteriores
  - Objetivo: população convergir em um valor ótimo de adaptação
  - Problema: convergência prematura devida a baixa diversidade



# Algoritmos genéticos

## ➤ População: características

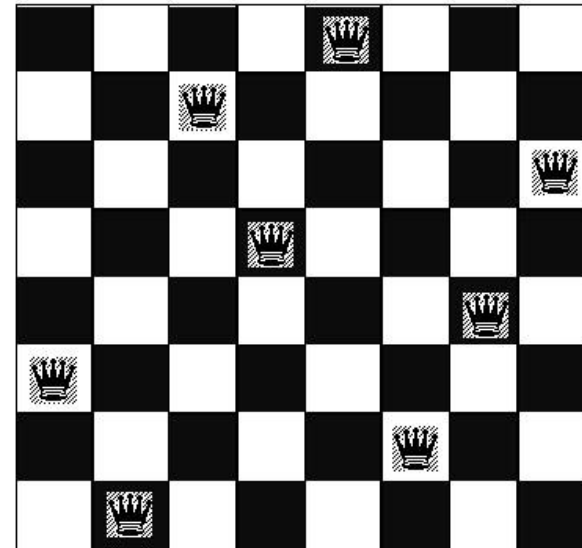
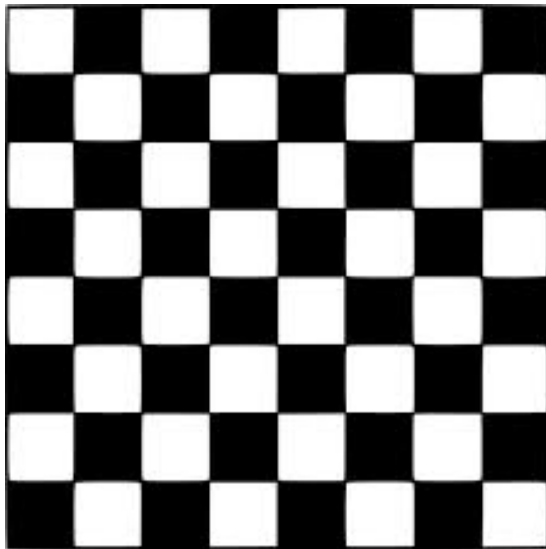
- **Diversidade:** mede o grau de variação entre os genótipos presentes na população (fundamental para a amplitude da busca)
- **Elite:** composta pelos indivíduos mais adaptados da população.
  - elitismo: os  $m$  melhores indivíduos (normalmente  $m = 1$ ) são mantidos a cada geração

# Algoritmo Genético

## Exemplos

# Algoritmos genéticos

## ➤ Exemplo1: Problema das 8 Rainhas



# Algoritmos genéticos

➤ Exemplo1: Problema das 8 Rainhas , solução.

2 4 7 4 8 5 5 2

3 2 7 5 2 4 1 1

2 4 4 1 5 1 2 4

3 2 5 4 3 2 1 3

- ✓ **População inicial:** Neste exemplo foram criados 4 indivíduos onde a posição no vetor corresponde à coluna do tabuleiro e o valor do algoritmo corresponde à linha (posicionamento de cada uma das 8 rainhas).

# Algoritmos genéticos

## ➤ Exemplo1: Problema das 8 Rainhas , solução.

2 4 7 4 8 5 5 2	24	31%
-----------------	----	-----

3 2 7 5 2 4 1 1	23	29%
-----------------	----	-----

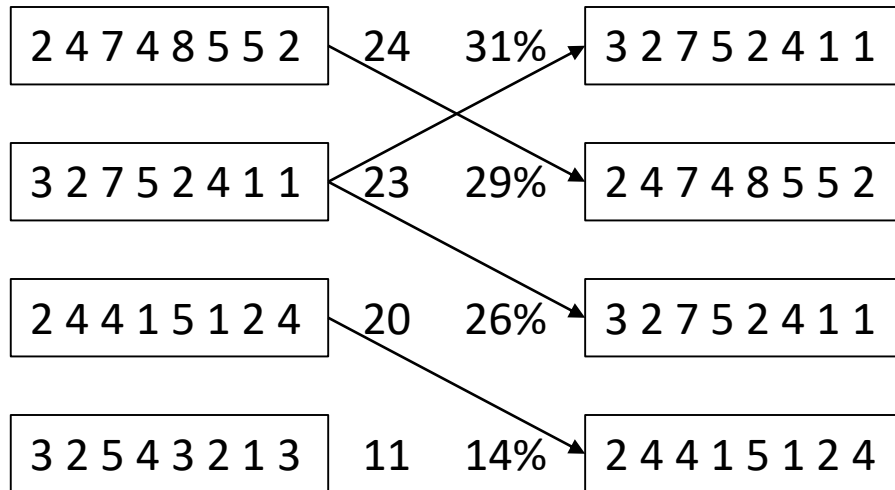
2 4 4 1 5 1 2 4	20	26%
-----------------	----	-----

3 2 5 4 3 2 1 3	11	14%
-----------------	----	-----

- ✓ **Função de avaliação:** Utiliza-se uma função que avalia quantos pares de rainha *não-atacantes* existem para cada indivíduo. O valor para a solução seria 28.

# Algoritmos genéticos

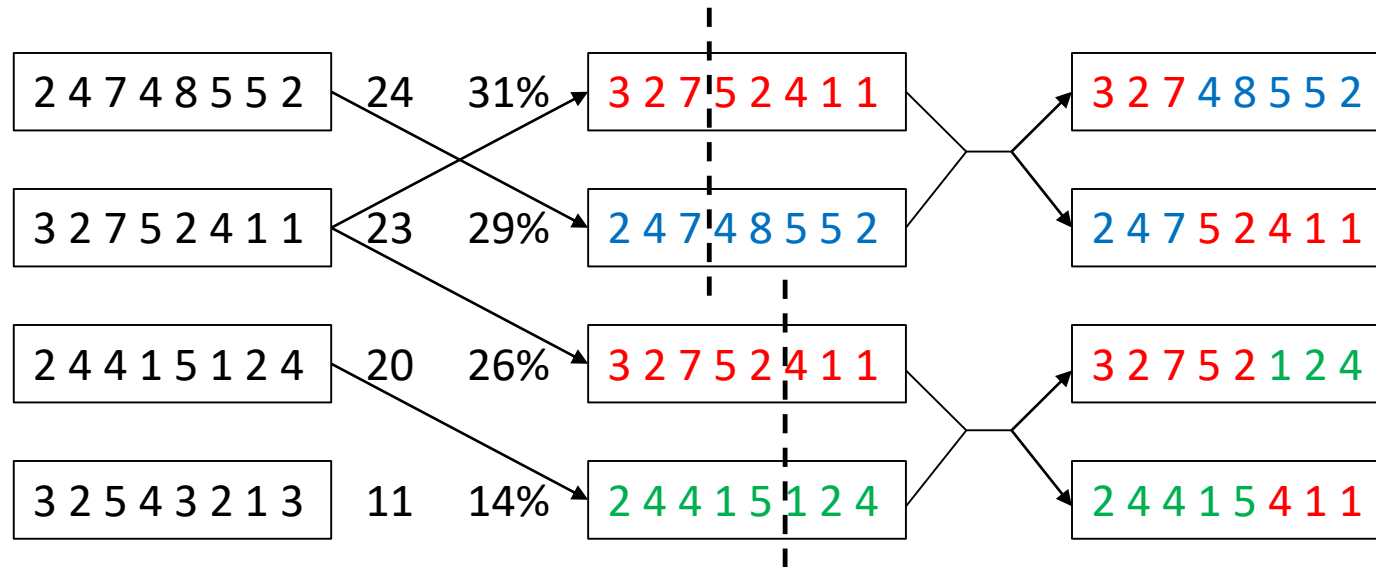
## ➤ Exemplo1: Problema das 8 Rainhas , solução.



- ✓ **Seleção:** Indivíduos são selecionados para reprodução com base na sua pontuação de fitness. Existem várias variantes para o critério de seleção.

# Algoritmos genéticos

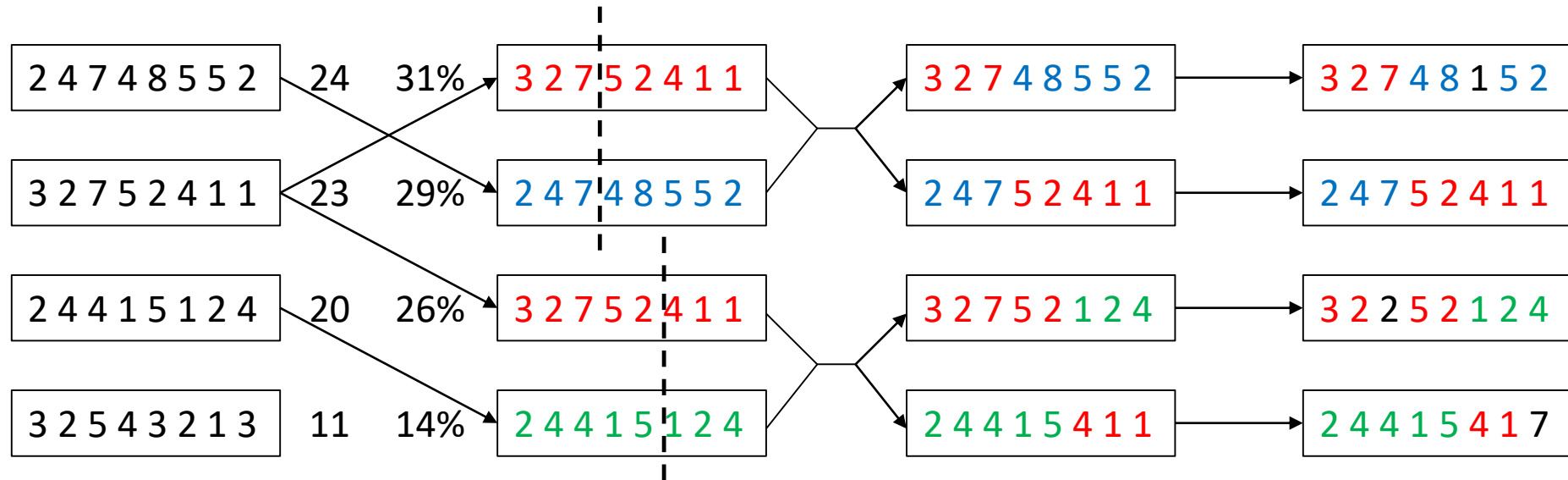
## ➤ Exemplo1: Problema das 8 Rainhas , solução.



- ✓ **Crossover:** Para cada par a ser cruzado é selecionado aleatoriamente um ponto de crossover dentre as posições na cadeia. Neste exemplo os pontos de crossover estão depois do terceiro dígito no primeiro par e depois do quinto dígito no segundo par.

# Algoritmos genéticos

## ➤ Exemplo1: Problema das 8 Rainhas , solução.



- ✓ **Mutação:** Cada posição está sujeita à mutação aleatória com uma pequena probabilidade independente.



# Algoritmos genéticos

## ➤ Exemplo2: Problema de otimização

- Maximizar a função  $f(x, y) = |x * y * \text{sen}(y^\pi/4)|$ , com  $x$  e  $y$  pertencentes ao intervalo  $[0,15]$ .

# Algoritmos genéticos

## ➤ Exemplo2: Problema de otimização, solução.

- Para evitar o valor de  $f(x, y) = 0$ , usaremos  $g(x, y) = 1 + f(x, y)$  como função de avaliação.
- Para o intervalo definido das variáveis, 4 bits para cada variável é suficiente. Logo, cada cromossomo será formado por 8 bits.
- Vamos utilizar uma taxa de mutação de 1%.
- Para fins explicativos, vamos manter uma população de 6 indivíduos.

# Algoritmos genéticos

## ➤ Exemplo2: Problema de otimização, solução.

Cromossomo	$x$	$y$	$g(x, y)$
0100 0011	4	3	9,5
0010 1001	2	9	13,7
1001 1011	9	11	71,0
0000 1111	0	15	1,0
1001 1001	5	5	18,7
1110 0011	14	3	30,7
Somatório das avaliações:			144,6

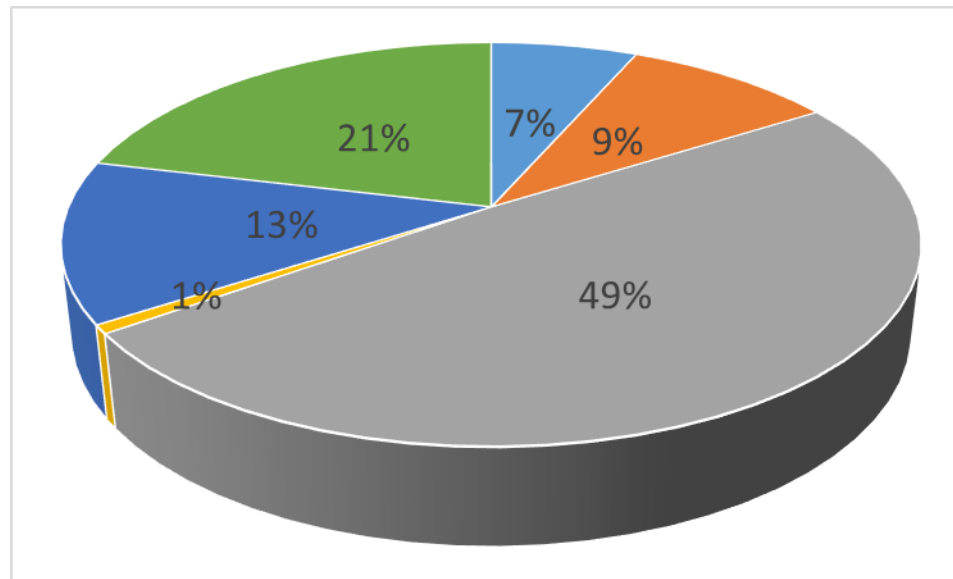
$$M_A = \frac{\sum_{i=1}^n f_o(i)}{n} = \frac{144,6}{6} = 24,1$$

- ✓ **População inicial:** Neste exemplo foram criados 6 indivíduos, onde os primeiros 4 bits representam o valor de  $x$  e os ultimos 4 bits o valor de  $y$ .
- ✓ **Função de avaliação:** Retorno da função  $g(x, y) = 1 + f(x, y)$

# Algoritmos genéticos

- A roleta para a população inicial

Cromossomo	$g(x, y)$	Intervalo	Aptidão (%)
0100 0011	9,5	$[0; 9,5[$	7%
0010 1001	13,7	$[9,5; 23,2[$	9%
1001 1011	71,0	$[23,2; 94,2[$	49%
0000 1111	1,0	$[94,2; 95,2[$	1%
1001 1001	18,7	$[95,2; 113,9[$	13%
1110 0011	30,7	$[113,9; 144,6[$	21%



# Algoritmos genéticos

➤ Exemplo2: Problema de otimização, solução.

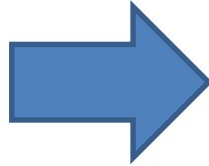
Números sorteados	Cromossomos escolhidos
12,8	00101001
65,3	10011011
108,3	10011001
85,3	10011011
1,8	01000011
119,5	11100011

✓ **Seleção:** Indivíduos são selecionados para reprodução com base na sua pontuação, de acordo com os intervalos definidos.

# Algoritmos genéticos

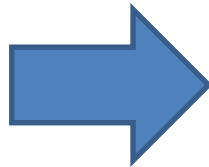
## ➤ Exemplo2: Problema de otimização, solução.

001 | 01001  
100 | 11011



00111011  
10001001

001010 | 01  
100110 | 11

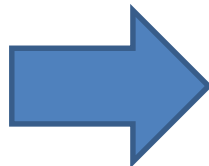


10011011  
10011001



Efeito da  
mutação

0100 | 0011  
1110 | 0011



01000011  
11100011

✓ **Crossover:** Para cada par a ser cruzado é selecionado aleatoriamente um ponto de crossover dentre as posições na cadeia.

# Algoritmos genéticos

➤ Exemplo2: Problema de otimização, solução.

Cromossomo	$x$	$y$	$g(x, y)$
00111011	3	11	24,3
10001001	8	9	51,9
10011011	9	11	71
10011000	9	8	1
01000011	4	3	9,5
11100011	14	3	30,7
Somatório das avaliações:			188,4

$$M_A = \frac{\sum_{i=1}^n f_o(i)}{n} = \frac{188,4}{6} = 47,1 > 24,1$$

✓ **Função de avaliação:** Avaliação da nova população utilizando a função  $g(x, y) = 1 + f(x, y)$

# Referências Bibliográficas

## Referência Bibliográfica Básica

[1] RUSSEL, S. e NORVIG, P. (2004). Inteligência Artificial. 2a. Edição. Editora Campos, 2004.

[2] PARSAYE, K.; CHIGNELL, M. Expert systems for experts. New Jersey: J. Wiley, 1988.

[3] NASCIMENTO JR., C. L.; YONEYAMA, T.; "Inteligência Artificial em Controle e Automação" - São Paulo: Blusher: FAPESP, 2004

[4] LUGER, George F. Inteligência Artificial. Porto Alegre: Ed. Bookman Companhia, 2004.

[5] LINDEN, Ricardo. Algoritmos Genéticos. Rio de Janeiro: Ed. Brasport, 2006.

[6] WOLPERT, D. H.; MACREADY, W. G., No Free Lunch Theorems for Search, Relatório Técnico SFI-TR-95-02-010, Santa Fe, Santa Fe, EUA, 1995.