

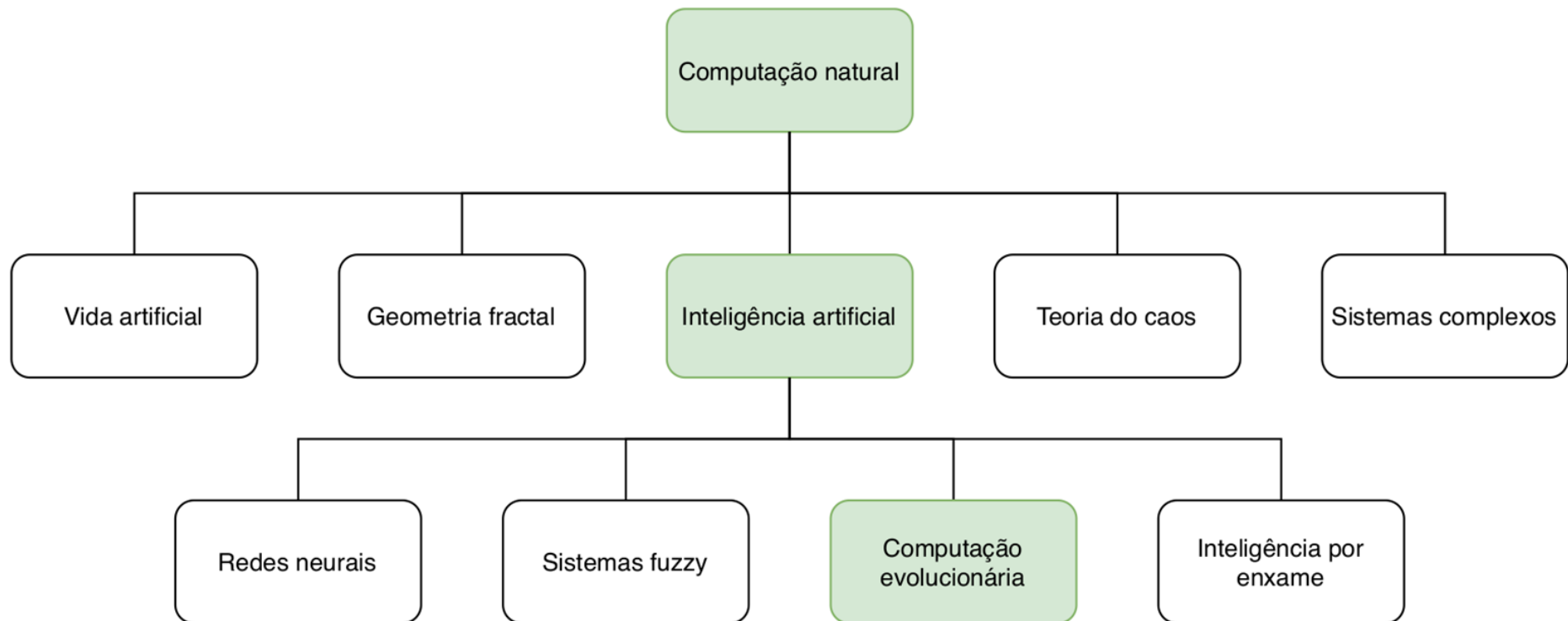
# Inteligência Artificial

## Além da Busca Clássica Parte 3

### Algoritmo Genético

**Prof. Jefferson Moraes**

# Algoritmo Genético



# Algoritmo Genético

- Fundamentado por John Henry Holland (1975)
- Popularizado por David E. Goldberg (1989)
- É uma estratégia bioinspirada utilizada para encontrar **soluções aproximadas** (ou ótimas) em problemas de otimização da engenharia e computação
- Baseia-se na biologia evolutiva (Charles Darwin & Gregor Mendel)
  - Seleção natural
  - Hereditariedade
  - Indivíduo, cromossomo, gene e alelo
  - Codificação
  - Recombinação
  - Mutação



# Algoritmo Genético

- **Termos chaves do AG**

- **Codificação**

- Significa representar computacionalmente um indivíduo
    - De maneira geral, pode ser: binária, inteira ou real

- **Indivíduo**

- É uma solução candidata do problema

- **População**

- É um conjunto de indivíduos

- **Função fitness**

- Avalia a qualidade das soluções candidatas
    - É a representação do problema que se quer otimizar

# Algoritmo Genético

- **Termos chaves do AG**

- **Seleção dos pais**

- Escolha de dois indivíduos da população atual

- **Operadores genéticos: gera novas soluções candidatas**

- **Cruzamento (crossover)**

- Troca de material genético entre os indivíduos selecionados, gerando filhos mais adaptados

- **Mutação**

- Alteração no gene do indivíduo que permite maior variabilidade genética na população

- **Seleção dos sobreviventes**

- Escolha dos indivíduos mais bem adaptados para a próxima geração

# Algoritmo Genético - Pseudocódigo

**P** ← inicializa\_população()

**FP** ← calcula\_fitness(**P**)

**While**(*critério de parada não é alcançado*)

**P'** ← selecao\_dos\_pais(**{P, FP}**)

**P''** ← cruzamento(**P'**)

**P'''** ← mutacao(**P''**)

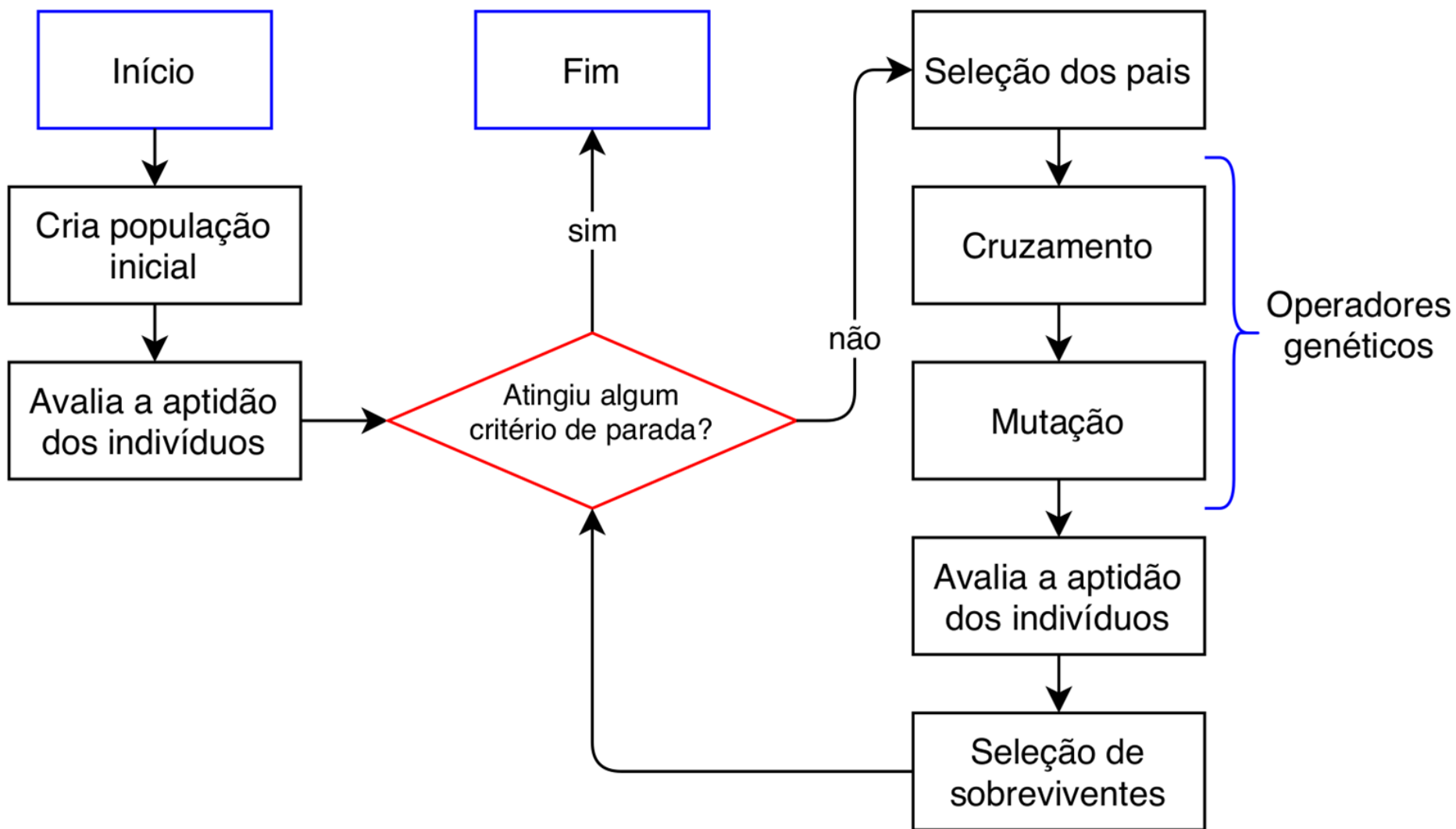
**FP'''** ← calcula\_fitness(**P'''**)

**{P, FP}** ← selecao\_dos\_sobreviventes(**{P, FP}, {P''', FP'''}**)

**End While**

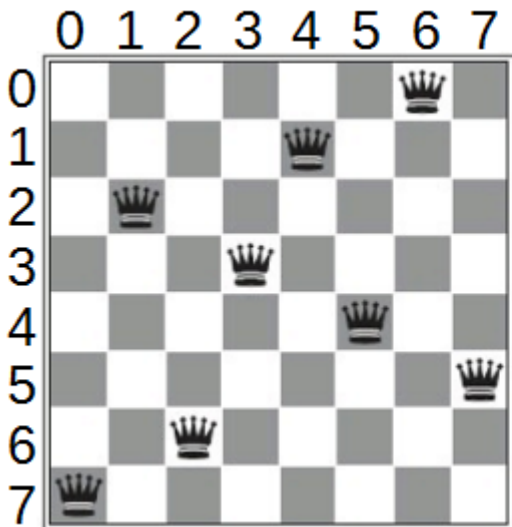
Retorna **melhor\_solucao(P)**

# Algoritmo Genético - Fluxograma



# Algoritmo Genético - Problema das 8 Rainhas

- **Codificação**: cada **indivíduo** é representado por um vetor de números inteiros com 8 posições
  - Cada posição do vetor representa uma coluna do tabuleiro
  - Cada valor de cada posição do vetor representa a linha da coluna onde a rainha está posicionada
- Exemplo de indivíduo



Codificação  
inteira

$X_i = [7 \ 2 \ 6 \ 3 \ 1 \ 4 \ 0 \ 5]$

Alternativamente:

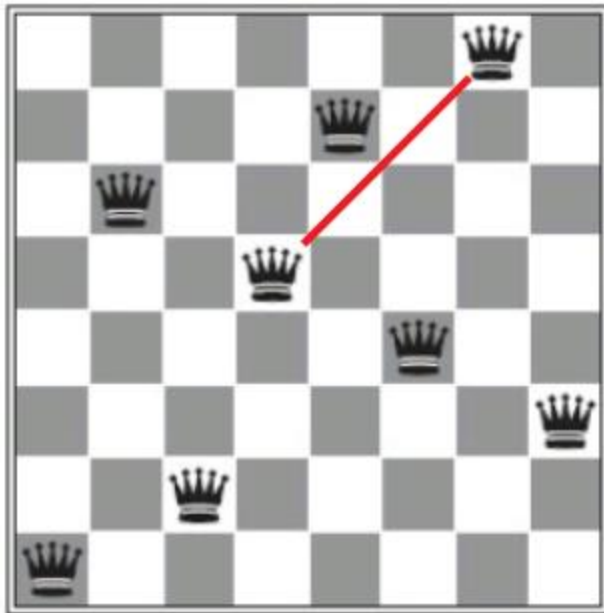
Codificação  
binária

$X_i = [111, 010, 110, 011, 001, 100, 000, 101]$



# Algoritmo Genético - Problema das 8 Rainhas

- **Função fitness:** deve retornar **valores melhores** para indivíduos melhores
- Exemplo de função fitness
  - Número de pares de rainhas **não** atacantes (maximização)
  - A solução que representa um ótimo global terá fitness igual a 28



$$X_i = [7 \ 2 \ 6 \ 3 \ 1 \ 4 \ 0 \ 5]$$

$$f(X_i) = 27$$

# Algoritmo Genético - Problema das 8 Rainhas

- Seleção dos pais

- Existem várias estratégias para selecionar indivíduos para o cruzamento
- Mais utilizadas
  - **Roleta**
    - A probabilidade de um indivíduo ser selecionado é proporcional ao seu valor de aptidão (fitness)
  - **Torneio**
    - Os indivíduos são selecionados aleatoriamente para um “ring”. O indivíduo com o melhor valor de aptidão é selecionado como pai

# Algoritmo Genético - Problema das 8 Rainhas

- Operador genético: cruzamento

- Troca de material genético entre dois pais com o objetivo de gerar filhos mais bem adaptados
- Algumas estratégias

- Ponto de corte

Pai 1:  $X_i = [7 \ 2 \ 6 \ | \ 3 \ 1 \ 4 \ 0 \ 5]$

Pai 2:  $X_j = [0 \ 1 \ 2 \ | \ 3 \ 4 \ 5 \ 6 \ 7]$

Randomiza o  
ponto de corte  
ex.: posição 3

Filho 1 =  $[7 \ 2 \ 6 \ 3 \ 4 \ 5 \ 6 \ 7]$

Filho 2 =  $[0 \ 1 \ 2 \ 3 \ 1 \ 4 \ 0 \ 5]$

- Uniforme

Pai 1:  $X_i = [7 \ 2 \ 6 \ 3 \ 1 \ 4 \ 0 \ 5]$

Pai 2:  $X_j = [0 \ 1 \ 2 \ 3 \ 4 \ 5 \ 6 \ 7]$

Gera um vetor binário  
ex.:  $[1 \ 1 \ 0 \ 1 \ 0 \ 0 \ 1 \ 0]$

Filho 1 =  $[0 \ 1 \ 6 \ 3 \ 1 \ 4 \ 6 \ 5]$

# Algoritmo Genético - Problema das 8 Rainhas

- Operador genético: mutação
  - Altera o material genético de um indivíduo
  - Algumas estratégias
    - Bit flip (apenas para codificação binária)

$X_i = [111, 010, 110, 0\mathbf{1}1, 001, 100, 000, 101]$

$X_i' = [111, 010, 110, 0\mathbf{0}1, 001, 100, 000, 101]$

Escolhe aleatoriamente uma posição para modificar o bit

- Permutação

$X_i = [7\ \mathbf{2}\ 6\ 3\ 1\ \mathbf{4}\ 0\ 5]$

$X_i' = [7\ \mathbf{4}\ 6\ 3\ 1\ \mathbf{2}\ 0\ 5]$

Escolhe aleatoriamente duas posições e permuta

# Algoritmo Genético - Problema das 8 Rainhas

- Seleção de sobreviventes

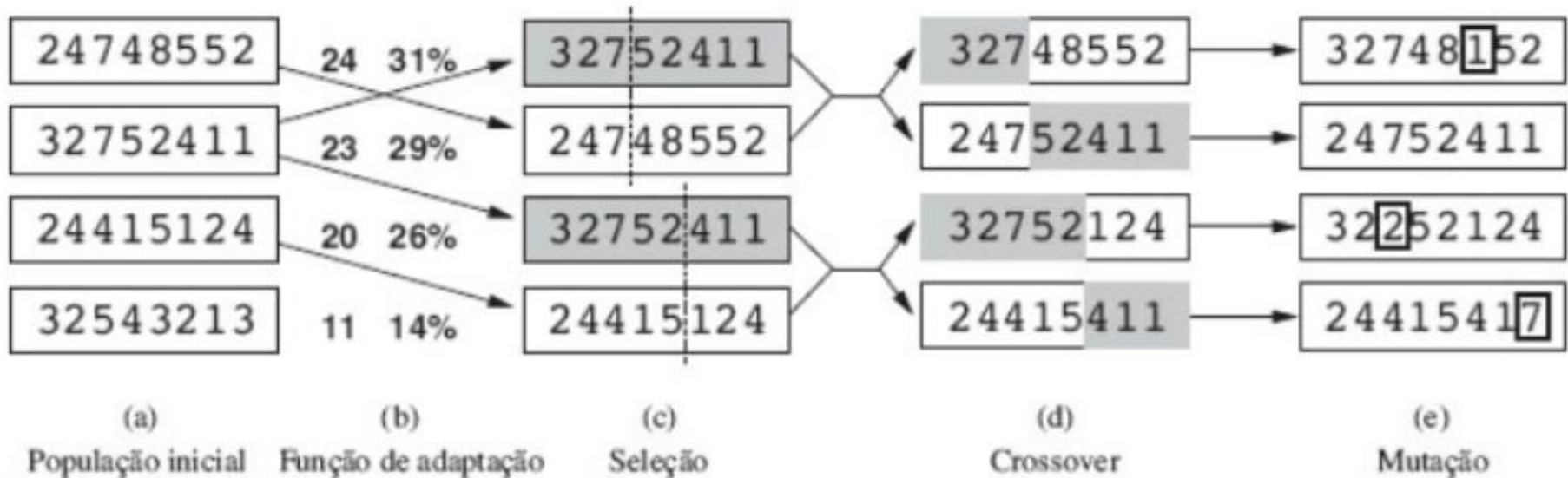
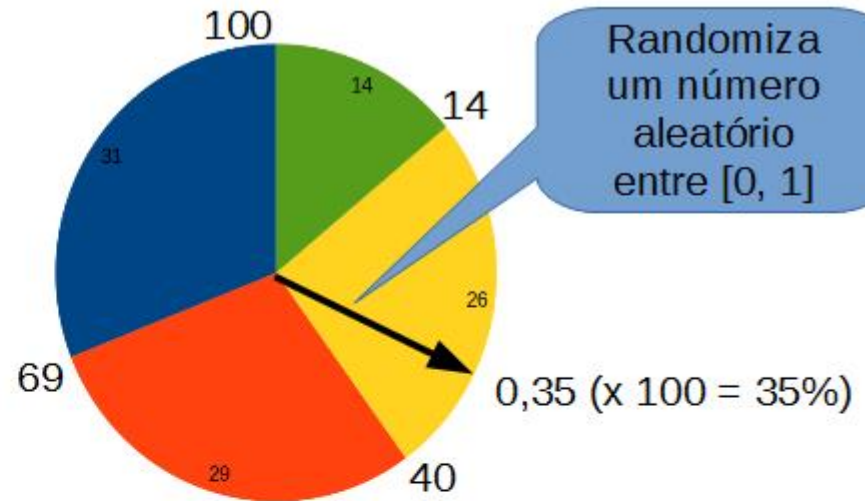
- Escolhe os indivíduos mais bem adaptados, entre a população atual e a população modificada após a mutação, para sobreviver para a próxima geração
- Algumas estratégias
  - **Troca de toda a população (geracional)**
    - Substitui toda a população atual pela população modificada após o operador de mutação
  - **Elitista com substituição total da população**
    - Substitui toda a população atual pela população modificada e preserva o melhor indivíduo
  - **Steady state**
    - Substitui parte da população (elimina os menos adaptados)

# Algoritmo Genético - Problema das 8 Rainhas

- **Parâmetros comuns em um AG**
  - **Tamanho da população:** número de soluções candidatas
  - **Taxa de cruzamento (TC)**
    - Se  $\text{rand} \leq \text{TC}$ , então ocorre cruzamento entre pais
    - Caso contrário, os pais se tornam filhos
  - **Taxa de mutação (TM)**
    - Se  $\text{rand} \leq \text{TM}$ , então ocorre mutação no indivíduo
    - Caso contrário, não ocorre
  - **Número de gerações:** número máximo de iterações

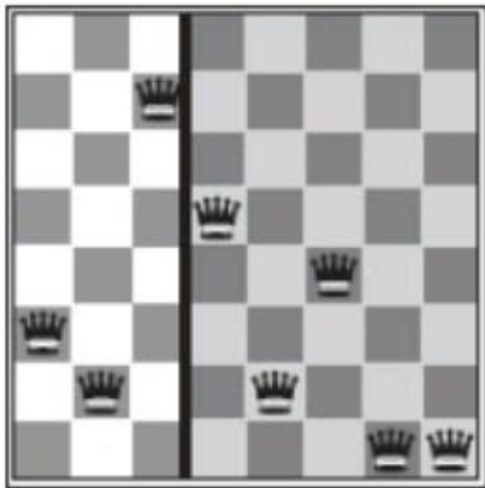
# Algoritmo Genético - Problema das 8 Rainhas

- Seleção dos pais: roleta
- Cruzamento: um ponto de corte
- Mutação: permutação



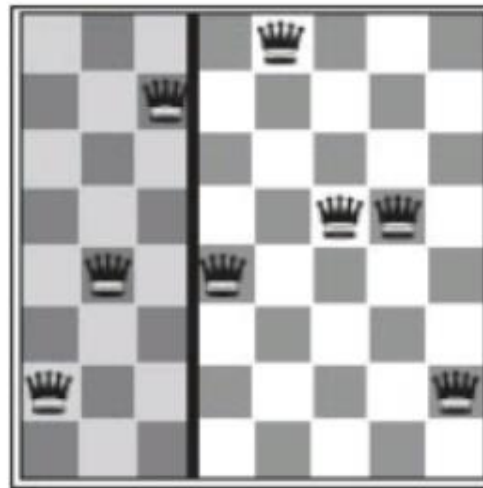
# Algoritmo Genético - Problema das 8 Rainhas

- Em(d), os próprios descendentes são criados por cruzamento das cadeias pais no ponto de crossover



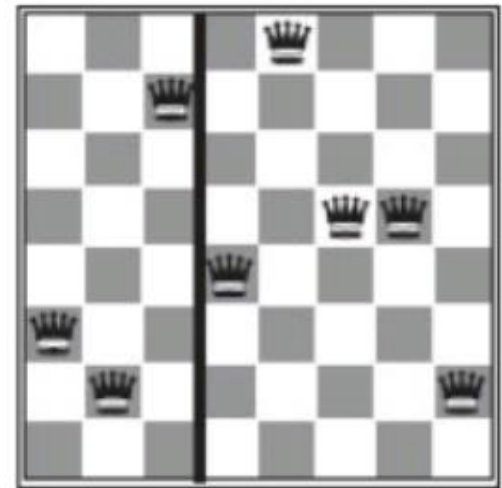
Pai 1: [5 6 1 | 3 5 4 7 7]

+



Pai 2: [6 4 1 | 4 0 3 3 6]

=



Filho 1: [5 6 1 | 4 0 3 3 6]



# Algoritmo Genético

- Os algoritmos genéticos combinam a **exploração aleatória** e a **troca de informações entre os indivíduos**
- De maneira geral, o operador de cruzamento realiza *exploration* e a mutação o *exploitation*
- **Exploration vs Exploitation**
  - Exploration (exploração)
    - Ocorre geralmente no início do algoritmo
    - Capacidade que o algoritmo tem de explorar o espaço de busca
  - Exploitation (intensificação)
    - Ocorre do meio para o final do algoritmo
    - Capacidade que o algoritmo tem de realizar um ajuste fino (fine tuning)