

# **ALGORITMOS GENÉTICOS**

Luís Morgado

ISEL-DEETC

# ALGORITMOS GENÉTICOS

## MOTIVAÇÃO

- **Os mecanismos naturais subjacentes parecem ser bastante robustos** e adequados ao suporte de mecanismos de adaptação, nomeadamente biológicos
- **Permitem pesquisar espaços de hipóteses complexos** e compostos por elementos interdependentes, cujo impacto global na selecção da hipótese mais adequada pode ser difícil de modelar
- **São facilmente paralelizáveis**, permitindo por isso tirar partido de sistemas computacionais massivamente paralelos

# ALGORITMOS GENÉTICOS COMO MÉTODOS DE PROCURA/OPTIMIZAÇÃO

## Diferenças em relação a outros métodos de procura

- **Utilizam uma codificação do conjunto de parâmetros** da função a otimizar, não os próprios parâmetros
- **Procura paralela** a partir de uma população de estados e não de um único estado
- **Paralelismo implícito** - cada indivíduo representa um conjunto variado de esquemas
- **Utilizam regras de transição de estado não determinísticas**
  - *Seleção* (reprodução)
  - *Cruzamento* (recombinação)
  - *Mutação*

# ALGORITMOS GENÉTICOS

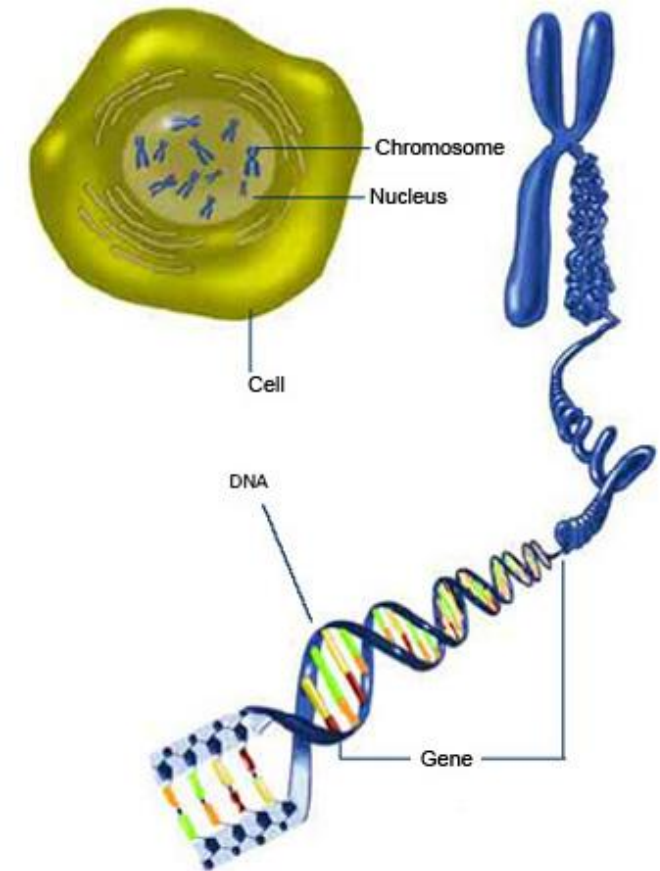
Algoritmos de procura baseados nos mecanismos biológicos de selecção natural e na genética, nomeadamente:

- Representação baseada em *cromossomas*
- Reprodução, cruzamento, mutação
- Sobrevivência dos mais adaptados
- Adaptação de sucessivas gerações ao ambiente
  - Surgimento de novas espécies

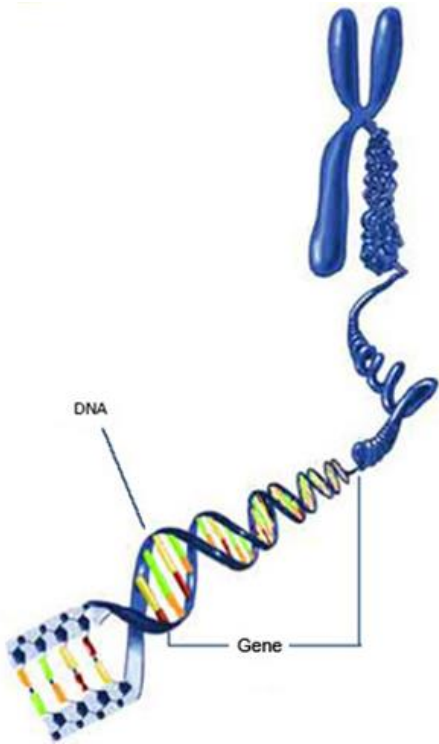
Origem: *Jonh Holland, Univ. Michigan, 1962*

Objectivos:

- Explicar o processo adaptativo dos sistemas naturais
- Desenvolver sistemas que incorporam modelos de mecanismos dos sistemas naturais



# TERMINOLOGIA NATURAL VS. ARTIFICIAL



**Cromossomas** .....

compostos por

**Genes**

caracterizados por

**Locus**

**Alelo**

**Genótipo** .....

(conteúdo genético)

**Fenótipo** .....

(organismo formado pela interação do  
genótipo com o ambiente)

**Cadeias de elementos simbólicos**

compostos por

**Elementos simbólicos**

caracterizados por

**Posição**

**Valor**

**Configuração codificada**

(cadeias de elementos simbólicos)

**Configuração decodificada**

(solução candidata)

# OPERADORES GENÉTICOS

## ***Seleção***

Processo através do qual indivíduos de uma geração são *seleccionados* para a próxima geração

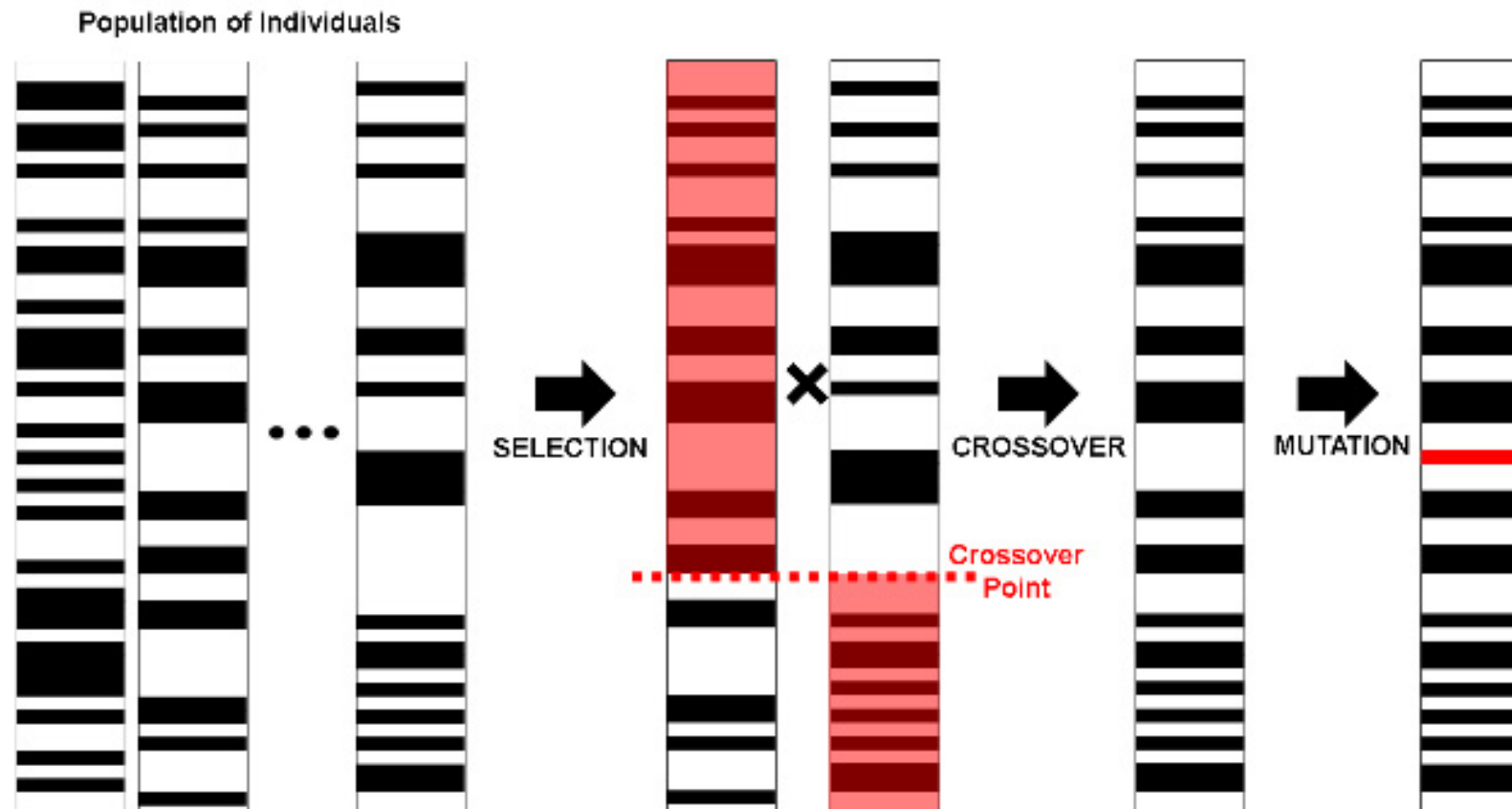
## ***Cruzamento***

Processo através do qual o material genético de um indivíduo é *misturado* com material genético de outro indivíduo

## ***Mutação***

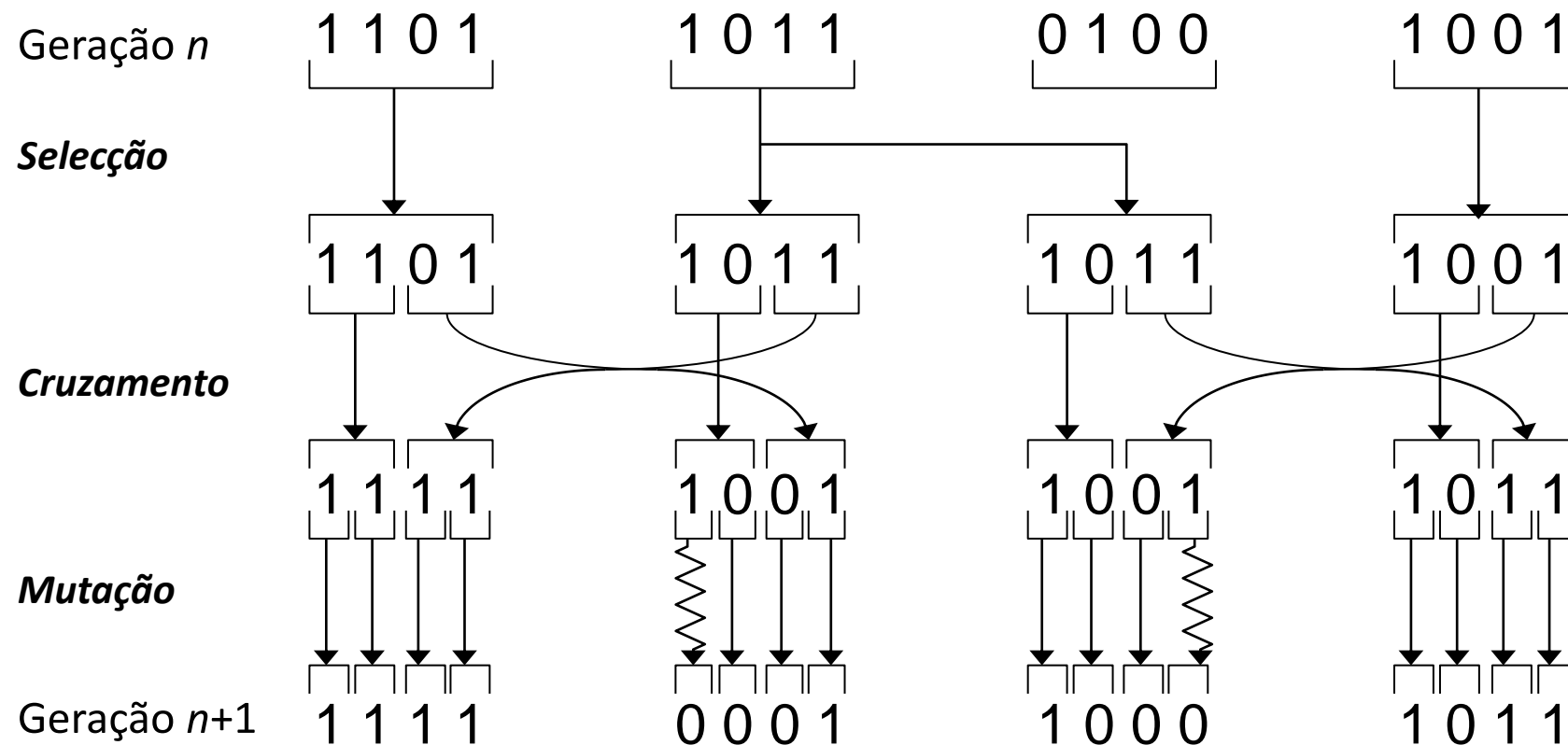
Processo através do qual o material genético é alterado

# OPERADORES GENÉTICOS



# EXEMPLO DE APLICAÇÃO DOS OPERADORES GENÉTICOS

## Cadeias binárias





# OPERADOR SELECÇÃO

## Por reprodução proporcional à adequação

A *função de adequação* ou *adaptação* (*fitness function*) é uma medida do valor de um indivíduo em relação a um determinado problema

Pretende-se que a selecção dos indivíduos para uma nova geração tenha em conta a sua *adequação*



Seja  $S$  uma população de indivíduos, a probabilidade de selecção de um indivíduo  $s \in S$  é proporcional à sua adequação normalizada, ou seja:

$$PSel(s) = \frac{f(s)}{\sum_{s_i \in S} f(s_i)}$$

# EXEMPLO

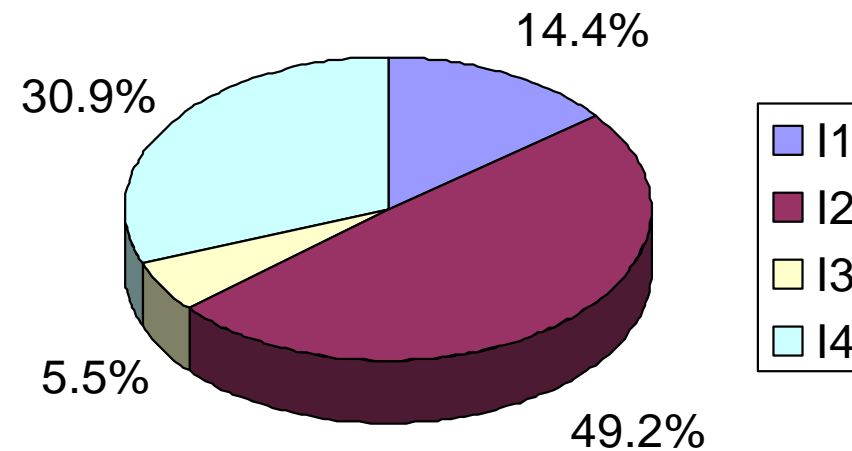
Considerando quatro indivíduos representados pelas seguintes *cadeias*:

10010, 11101, 01000, 10011

	$s$	$f(s)$	$PSel(s)$
$I1$	10010	169	0.14
$I2$	11101	576	0.49
$I3$	01000	64	0.05
$I4$	10011	361	0.31
Total		1170	1.00

# SELECÇÃO POR *ROLETA*

Probabilidade de selecção de um indivíduo



$n$  jogadas da roleta permitem seleccionar  $n$  novos indivíduos (*cadeias*)

A probabilidade de selecção de um indivíduo  $s$  é  $PSel(s)$

# EXEMPLO

$n$		$s$	$f(s)$	$PSel(s)$	$n*PSel(s)$	$N^o \text{ de cópias}$
4	$I1$	10010	169	0.14	0.58	1
	$I2$	11101	576	0.49	1.97	2
	$I3$	01000	64	0.05	0.22	0
	$I4$	10011	361	0.31	1.23	1

Número de cópias esperado para uma cadeia  $s$ :

$$n \cdot PSel(s) = f(s) \cdot \frac{n}{\sum_{Si \in S} f(Si)} = \frac{f(s)}{\overline{f}}$$

# MÉTODOS DE SELECÇÃO

- Devido a ser um processo estocástico, a selecção por *roleta* permite que **alguns indivíduos sejam seleccionados um número de vezes superior ao correspondente à sua adequação**
- $PSel(s)$  é um valor real, ao passo que o número de indivíduos seleccionados é um valor inteiro
- Existem outros métodos de selecção, por exemplo ***Seleção por categoria*** (*Rank selection*)
  - Utilizado quando os indivíduos têm valores de adequação muito próximos
    - Para evitar perda de pressão selectiva para aumento da adequação

# OPERADOR CRUZAMENTO

O processo de selecção produz um conjunto de candidatos a cruzamento (*mating pool*)

Cruzamento de uma cadeia  $s$

- Tendo em conta uma probabilidade de cruzamento  $P_c$ 
  - Obter aleatoriamente uma outra cadeia para constituição de um par
  - Sendo  $l$  o comprimento das cadeias, determinar um inteiro  $c \in [1, l-1]$ 
    - $c$  representa a posição de cruzamento

# EXEMPLO

Par de cadeias a *cruzar*:

1 1 0 1 1 0 1

0 1 1 1 0 1 0

$l = 7, c = 4$

1 1 0 1 | 1 0 1  
0 1 1 1 | 0 1 0

Posição de cruzamento

Troca de material genético

1 1 0 1 **0** 1 0  
0 1 1 1 **1** 0 1

# OPERADOR CRUZAMENTO

## Outras formas de cruzamento

- ***Cruzamento em dois pontos***: Similar ao anterior, só que neste caso são seleccionados duas posições de cruzamento distintas
- ***Cruzamento multi-ponto***: Extensão do anterior para diversas posições de cruzamento
- ***Cruzamento uniforme***: Cada novo gene é gerado por cópia do gene correspondente de um dos progenitores, de acordo com uma *máscara de cruzamento* gerada aleatoriamente, a qual indica qual dos progenitores deve fornecer o gene

**Máscara de cruzamento** :     **1 0 1 0 0 1 0 1**

Progenitor 1 :     1 1 0 1 0 0 1 0

↓     ↓         ↓     ↓

**Descendente 1** :     1 1 0 1 0 0 0 0

↑     ↑     ↑     ↑

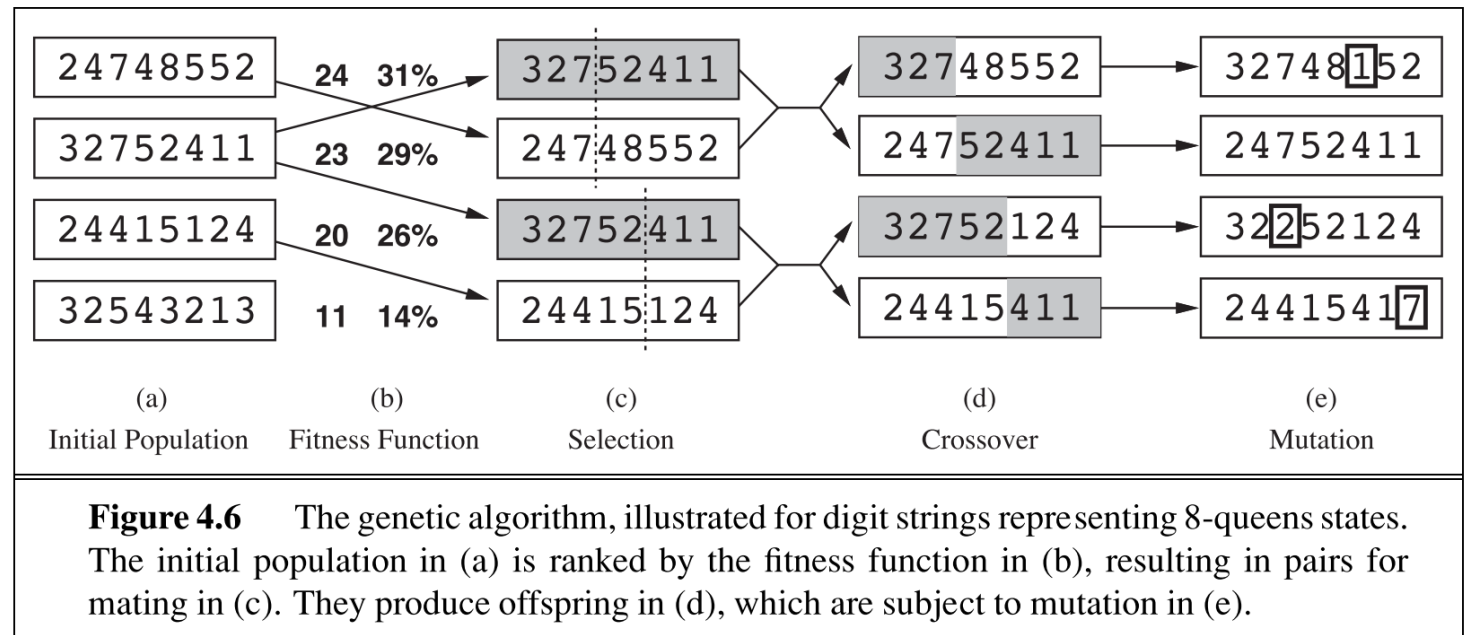
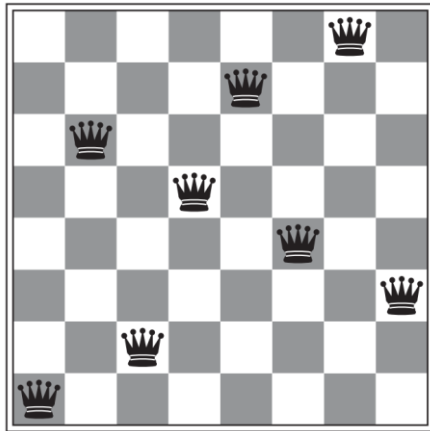
Progenitor 2 :     0 1 1 1 0 1 0 0



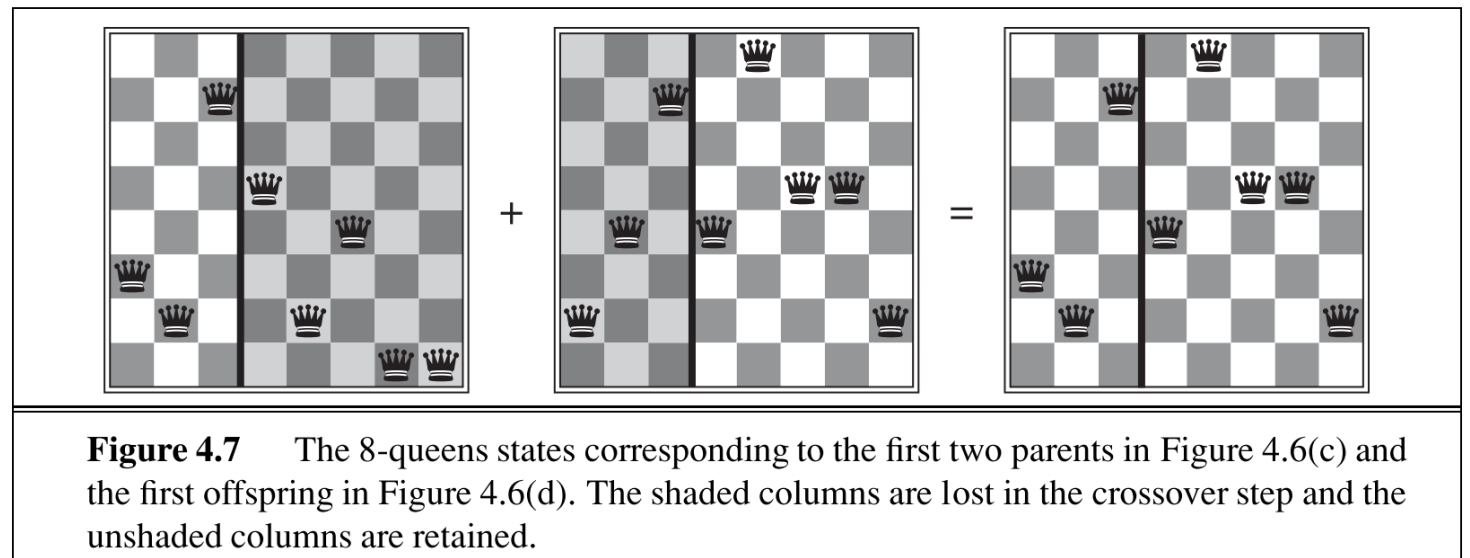
# OPERADOR MUTAÇÃO

- Alteração ocasional (com pequena probabilidade) de um valor numa das posições de uma cadeia
- Pode servir de precaução contra a perda de material genético potencialmente útil
- É uma operação não fundamental
- **Deve ser relativamente baixa**

# OPERADORES GENÉTICOS: EXEMPLO



**Figure 4.6** The genetic algorithm, illustrated for digit strings representing 8-queens states. The initial population in (a) is ranked by the fitness function in (b), resulting in pairs for mating in (c). They produce offspring in (d), which are subject to mutation in (e).

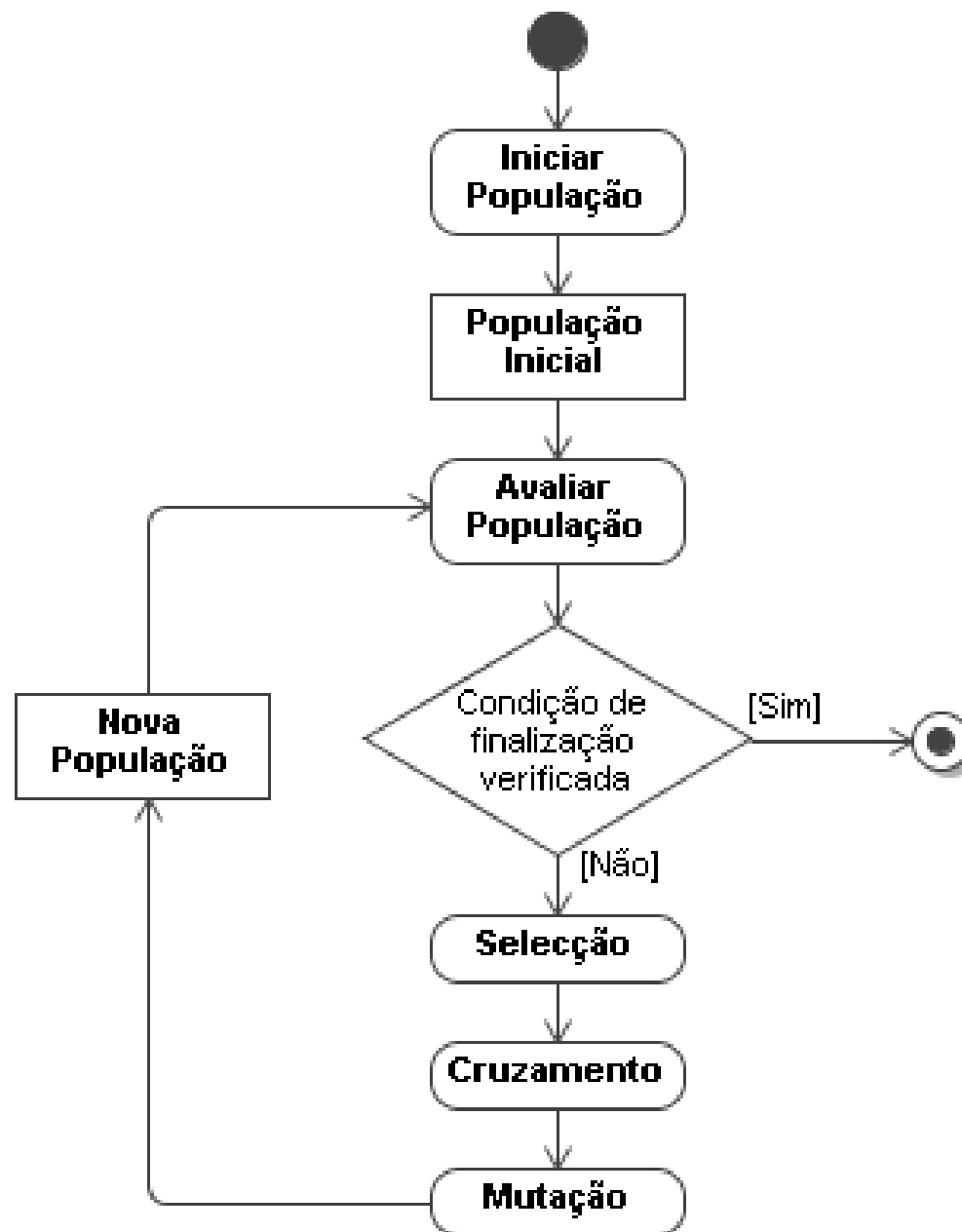


**Figure 4.7** The 8-queens states corresponding to the first two parents in Figure 4.6(c) and the first offspring in Figure 4.6(d). The shaded columns are lost in the crossover step and the unshaded columns are retained.

# POPULAÇÃO

- **Conjunto de soluções candidatas**
  - Indivíduos - *fenótipo*
  - Representadas por cromossomas - *genótipo*
- **Geração**
  - População no instante  $t + 1$ , constituída por indivíduos resultantes da aplicação das operações *selecção* (*reprodução*), *cruzamento* e *mutação* à população no instante  $t$
- **Evolui com o tempo**

# ALGORITMO GENÉTICO



# ALGORITMO GENÉTICO: EJEMPLO

---

```
function GENETIC-ALGORITHM(population, fitness) returns an individual
  repeat
    weights  $\leftarrow$  WEIGHTED-BY(population, fitness)
    population2  $\leftarrow$  empty list
    for i = 1 to SIZE(population) do
      parent1, parent2  $\leftarrow$  WEIGHTED-RANDOM-CHOICES(population, weights, 2)
      child  $\leftarrow$  REPRODUCE(parent1, parent2)
      if (small random probability) then child  $\leftarrow$  MUTATE(child)
      add child to population2
    population  $\leftarrow$  population2
  until some individual is fit enough, or enough time has elapsed
  return the best individual in population, according to fitness

function REPRODUCE(parent1, parent2) returns an individual
  n  $\leftarrow$  LENGTH(parent1)
  c  $\leftarrow$  random number from 1 to n
  return APPEND(SUBSTRING(parent1, 1, c), SUBSTRING(parent2, c + 1, n))
```

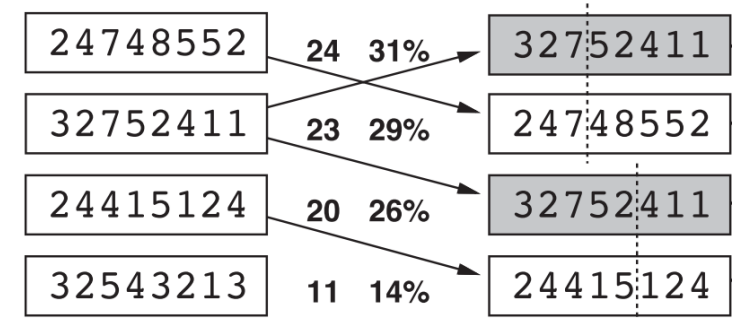
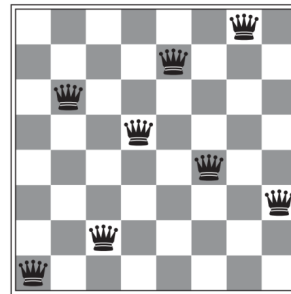
**Figure 4.7** A genetic algorithm. Within the function, *population* is an ordered list of individuals, *weights* is a list of corresponding fitness values for each individual, and *fitness* is a function to compute these values.

# REPRESENTAÇÃO DO PROBLEMA

## ESCOLHA DE UMA CODIFICAÇÃO

### Importância da representação

- Garantir restrições do problema
- Reduzir multiplicidade
- **Exemplo**
  - Problema das N-Rainhas



### Princípios base

- Escolher o **menor alfabeto** que permite uma **representação natural** do problema
- Escolher a codificação de tal forma que os **elementos simbólicos sejam significativos para o problema em análise e independentes de outros elementos**

### Problemas a ter em conta

- Pendor de procura (*search bias*) eventualmente não adequado
- Semântica posicional

# TÉCNICAS BASE DE CODIFICAÇÃO

**Lista de características: sequência de bits**



Cada bit representa uma única propriedade, característica, ou predicado.

bit  $i = 1$  : Característica  $i$  presente

bit  $i = 0$  : Característica  $i$  ausente

***Linguagem de descrição de conceitos correspondente:***

1 - Característica presente nas instâncias

0 - Característica não presente nas instâncias

# - Característica com valores inconsistentes para instâncias distintas

## **Aplicação**

Classes cuja descrição possa ser realizada com base num conjunto de características essenciais, cuja presença ou ausência seja obrigatória

# TÉCNICAS BASE DE CODIFICAÇÃO

## Ordenação linear

- Cadeia de elementos utilizada para representar parâmetros pertencentes a um conjunto ordenado de valores mutuamente exclusivos
  - Valores inteiros
  - Parâmetros contínuos com uma determinada precisão
  - Valores simbólicos

## Exemplo

**Problema:** Encontrar a configuração de pesos de uma rede neuronal

**Codificação:** Blocos com valores reais representam pesos das ligações

**Cromossoma:** [0.3254, -2.5310, 0.5287, 1.4629, 0.7461]



# REPRESENTAÇÃO DO PROBLEMA

- **Técnicas de codificação**
  - **Codificação binária**
    - Sequência de bits
  - **Codificação inteira**
    - Atributos inteiros ordenados ou não ordenados
  - **Codificação real**
    - Aproximação de valores reais
  - **Permutações**
    - Para problemas que implicam restrições de ordem
    - Valores não podem ser repetidos no mesmo genótipo
  - **Codificação em árvore**
    - Representação de árvores sintáticas (e.g. programa)
    - Programação genética

# **EXEMPLO: Problema do Caixeiro Viajante**

## **Descrição do problema**

- **Informação a representar**
  - Cidades e respectivas distâncias entre cidades
- **Resultado a obter**
  - Percurso tal que:
    - Cada cidade seja visitada uma única vez
    - Termine no ponto de partida
    - Minimize a distância total percorrida

# EXEMPLO: Problema do Caixeiro Viajante

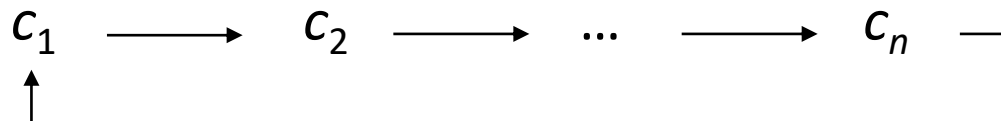
## Codificação dos percursos

### Codificação 1: *Codificação binária e operadores genéticos base*

- Possível repetição de cidades num percurso - *percursos inválidos*
- Possível obtenção de códigos que não representam qualquer cidade

### Codificação 2: *Codificação através de um vector de inteiros*

- Necessidade de operadores genéticos específicos
- O vector  $(c_1, c_2, \dots, c_n)$  representa o percurso



# EXEMPLO: Problema do Caixeiro Viajante

## Operadores de cruzamento

### Cruzamento com preservação de ordem

Escolha aleatória de dois pontos de corte que determinam um sub-percurso a copiar:

Percurso 1 :	(1 2 3   4 5 6   7 8 9)	→	Percurso 1 <sub>d</sub> :	( _ _ _ 6 9 4 _ _ _ )
Percurso 2 :	(5 8 7   6 9 4   3 2 1)		Percurso 2 <sub>d</sub> :	( _ _ _ 4 5 6 _ _ _ )

**Preservação da ordem das cidades restantes:**

- **Obtenção da ordem das cidades a partir do segundo ponto de corte**

Percurso 1: 7 → 8 → 9 → 1 → 2 → 3 → 4 → 5 → 6

- **Eliminação das cidades já visitadas**

Percurso 1: 7 → 8 → 9 → 1 → 2 → 3 → 4 → 5 → 6

- **Colocação das cidades restantes a partir do segundo ponto de corte**

Percurso 1 :	( _ _ _ 6 9 4 _ _ _ )	→	Percurso 1 <sub>d</sub> :	( 2 3 5 6 9 4 7 8 1 )
Percurso 2 :	( _ _ _ 4 5 6 _ _ _ )		Percurso 2 <sub>d</sub> :	( 8 7 9 4 5 6 3 2 1 )

# EVOLUÇÃO DA POPULAÇÃO

Como caracterizar a evolução de uma população no âmbito de um algoritmo genético ?

Indivíduo  $\longleftrightarrow$  Cadeia

Indivíduos com  
*semelhanças*  $\longleftrightarrow$  Esquema

Um *esquema* descreve um conjunto de cadeias

# ESQUEMAS

Representam **padrões de semelhança** entre cadeias

Permitem relacionar as semelhanças entre cadeias com altos valores de adequação

Cada esquema representa várias cadeias

Cada cadeia corresponde a múltiplos esquemas

## **Alfabeto para construção de esquemas:**

Alfabeto de cadeias:  $V$

Alfabeto de esquemas:  $V \cup \{*\}$

## **Exemplo:**

$V = \{0, 1\}$

$V \cup \{*\} = \{0, 1, *\}$

Esquema  $H = 0^*00^*1^{**}1$  (Representa várias cadeias)

# CARACTERIZAÇÃO DE UM ESQUEMA

**Ordem** de um esquema  $H$ :  $o(H)$

*Número de posições fixas do esquema*

Exemplo:  $H = 1****$        $o(H) = 1$

$H = 10*1*$        $o(H) = 3$

**Comprimento de definição** de um esquema:  $\delta(H)$

*Distância máxima entre posições fixas*

Exemplo:  $H = 10*1*$        $\delta(H) = 3$

$H = **11*$        $\delta(H) = 1$

$H = 1****$        $\delta(H) = 0$

# ESQUEMAS

Como são os esquemas afectados pela aplicação de operadores?

- **Seleccção**
  - As cópias de uma cadeia representam os mesmos esquemas
- **Cruzamento**
  - Quanto menor for o comprimento de definição  $\delta$  menor será a probabilidade de o cruzamento fazer desaparecer o esquema
- **Mutação**
  - Quanto menor for a ordem  $o$  menor será a probabilidade de a mutação fazer desaparecer o esquema



# EQUAÇÃO DE CRESCIMENTO DE UMA POPULAÇÃO

Considerando apenas a reprodução

$$\underbrace{m(H, t+1)}_{\substack{\uparrow \\ \text{Número de representantes} \\ \text{de H no instante } t+1}} = m(H, t) \cdot \frac{f(H)}{\bar{f}}$$

Probabilidade de sobrevivência de um esquema a um cruzamento

$$p_{sc}(H) \geq 1 - \underbrace{p_c}_{\substack{\nearrow \\ \text{Probabilidade de cruzamento}}} \cdot \underbrace{\frac{\delta(H)}{l-1}}_{\substack{\longleftarrow \\ \text{Probabilidade máxima de} \\ \text{destruição em caso de} \\ \text{cruzamento}}}$$

# EQUAÇÃO DE CRESCIMENTO DE UMA POPULAÇÃO

Probabilidade de sobrevivência de um esquema a uma mutação:

$$(1 - p_m)^{o(H)}$$



Probabilidade de  
mutação de uma posição

Sendo  $p_m \ll 1$ :

$$(1 - p_m)^{o(H)} \approx 1 - o(H) \cdot p_m$$

Equação de crescimento geral (***Schema theorem***):

$$m(H, t + 1) \geq m(H, t) \cdot \frac{f(H)}{\bar{f}} \left[ 1 - p_c \cdot \frac{\delta(H)}{l - 1} - o(H) \cdot p_m \right]$$

[Holland, 1975]

# EQUAÇÃO DE CRESCIMENTO DE UMA POPULAÇÃO

A equação de crescimento geral (*Schema theorem*) permite concluir que **esquemas com maior adequação vão aumentando o seu número de representantes em gerações sucessivas** (de forma exponencial).

**Cada indivíduo representa um conjunto variado de esquemas, assim sendo o número de esquemas efectivamente processado é superior ao número de indivíduos ( $n$ ) da população.**



**Paralelismo implícito**

# BLOCOS CONSTRUTORES (*BUILDING BLOCKS*)

Esquemas com **alta adequação média**, pequeno comprimento de definição e pequena ordem

Em termos gerais os algoritmos genéticos

**Descobrem**

**Reforçam**

**Recombinam**

***blocos construtores*** de soluções de uma forma massivamente paralela

# REFERÊNCIAS

[Russel & Norvig, 2022]

S. Russell, P. Norvig, *Artificial Intelligence: A Modern Approach*, 4th Edition, Pearson, 2022

[Russel & Norvig, 1995]

S. Russell, P. Norvig, *Artificial Intelligence: A Modern Approach*, 3rd Edition, Prentice Hall, 2010

[Goldberg , 1989]

David E. Goldberg, "Genetic Algorithms", *Addison-Wesley*, 1989.

[Michalewicz, 1994]

Zbigniew Michalewicz, "Genetic Algorithms + Data Structures = Evolution Programs", *Springer Verlag*, 1994.

[Mitchell, 1996]

Melanie Mitchell, "An Introduction to Genetic Algorithms", *MIT Press*, 1996.

[Booker, 1991]

Lashon B. Booker, "Representing Attribute-Based Concepts in a Classifier System", in "*Foundations of Genetic Algorithms*", *Morgan Kaufmann*, 1991.

[Buckles & Petry , 1992]

B. Buckles, F. Petry, "Genetic Algorithms", *IEEE Press*, 1992.

[Mitchell, 1997]

Tom M. Mitchell, "Machine Learning", *McGraw-Hill*, 1997.

[Hornby *et al.*, 2006]

G. Hornby, A. Globus, D. Linden, J. Lohn, "Automated antenna design with evolutionary algorithms". American Institute of Aeronautics and Astronautics, 2006

[Holland, 1975]

J. Holland, "Adaptation in Natural and Artificial Systems". MIT Press, 1975