

# Inteligência Artificial

---

Busca Local: Algoritmos Genéticos

Prof. Dr<sup>a</sup>. Andreza Sartori

[asartori@furb.br](mailto:asartori@furb.br)

# Documentos Consultados/Recomendados

- ARTERO, Almir Olivette. **Inteligência artificial: teórica e prática**. 1. ed. São Paulo: Livraria da Física, 2008.
- CARR, Jenna. **An Introduction to Genetic Algorithms**, 2014.
- CRUZ, Adriano Joaquim de Oliveira. **Inteligência Computacional: Algoritmos Genéticos**. UFRJ, 2013.
- KLEIN, Dan; ABBEEL, Pieter. **Intro to AI**. UC Berkeley. Disponível em: <http://ai.berkeley.edu> Acesso em: 31 jul. 2016.
- LIMA, Edirlei Soares. **Inteligência Artificial**. PUC-Rio, 2015.
- RUSSELL, Stuart J. (Stuart Jonathan); NORVIG, Peter. **Inteligência artificial**. Rio de Janeiro: Campus, 2013. 1021 p, il.

# Conteúdo Programático

**Unidade 1:** Fundamentos de Inteligência Artificial

**Unidade 2:** Busca

**Unidade 3:** Sistemas Baseados em Conhecimento

**Unidade 4:** Redes Neurais Artificiais

**Unidade 5:** Aplicações de Inteligência Artificial



# Conteúdo Programático

**Unidade 1:** Fundamentos de Inteligência Artificial

**Unidade 2:** Busca

**Unidade 3:** Sistemas Baseados em Conhecimento

**Unidade 4:** Redes Neurais Artificiais

**Unidade 5:** Aplicações de Inteligência Artificial



# Conteúdo Programático

**Unidade 1:** Fundamentos de Inteligência Artificial

**Unidade 2:** Busca

2.1. Resolução de Problemas por meio de busca

2.2. Busca Cega ou Exaustiva

2.3. Busca Heurística

2.4. Busca Competitiva

**2.5. Busca Local**

2.5.1 Algoritmos Genéticos (AG)



# Métodos de Busca

- **Busca Cega ou Exaustiva:**

- Não tem nenhuma informação adicional sobre os estados, isto é, não sabe qual o melhor nó da fronteira a ser expandido. Apenas distingue o estado objetivo dos não objetivos.

- **Busca Heurística:**

- Ou busca com informação, estima qual o melhor nó da fronteira a ser expandido baseado em funções heurísticas.

- **Busca Competitiva:**

- Considera que há oponentes hostis e imprevisíveis. Ex: Jogos

- **Busca Local:**

- Operam em um único estado e movem-se para a vizinhança deste estado.
- **Algoritmos Genéticos:**
  - Variante de Busca Local em que é mantida uma grande população de estados. Novos estados são gerados por mutação e por crossover, que combina pares de estados da população.

# Métodos de Busca

- **Busca Cega ou Exaustiva:**

- Não tem nenhuma informação adicional sobre os estados, isto é, não sabe qual o melhor nó da fronteira a ser expandido. Apenas distingue o estado objetivo dos não objetivos.

- **Busca Heurística:**

- Ou busca com informação, estima qual o melhor nó da fronteira a ser expandido baseado em funções heurísticas.

- **Busca Competitiva:**

- Considera que há oponentes hostis e imprevisíveis. Ex: Jogos

- **Busca Local:**

- Operam em um único estado e movem-se para a vizinhança deste estado.
- **Algoritmos Genéticos:**
  - Variante de Busca Local em que é mantida uma grande população de estados. Novos estados são gerados por mutação e por crossover, que combina pares de estados da população.

# Busca Local

- Em muitos problemas o **caminho para a solução é irrelevante.**
  - **Jogo das n-rainhas:** o que importa é a configuração final e não a ordem em que as rainhas foram posicionadas.
  - **Outros exemplos:**
    - Projeto de Circuitos eletrônicos;
    - Layout de instalações industriais;
    - Escalonamento de jornadas de trabalho;
    - Otimização de redes de telecomunicações.
- Se o caminho para a solução não importa, podemos utilizar um algoritmo de **busca local.**



# Busca Local

- Algoritmos de busca local operam sobre um **único estado corrente**, ao invés de vários caminhos.
- Em geral se movem apenas para os **vizinhos** desse estado.
- Normalmente, o caminho seguido pelo algoritmo não é guardado.

# Busca Local

- **Vantagens:**

- Ocupam pouquíssima memória (normalmente constante).
- Podem encontrar soluções razoáveis em grandes ou infinitos espaços de estados.

- **São úteis para resolver problemas de otimização.**

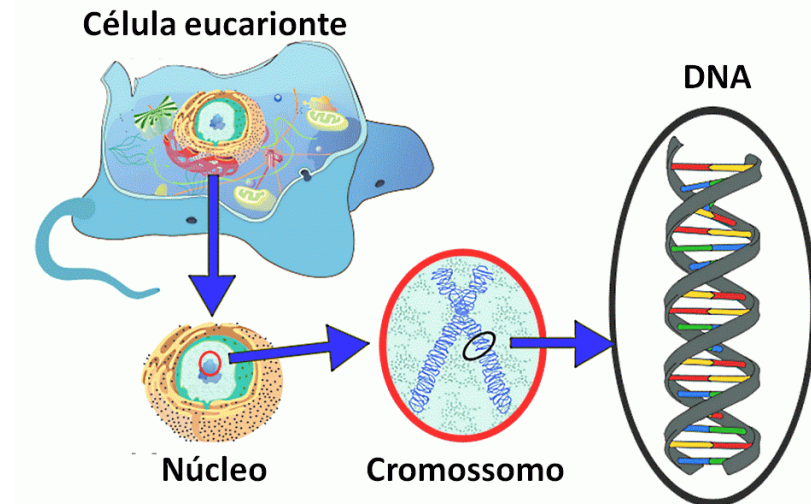
- Nos quais o objetivo é encontrar o melhor estado de acordo com uma **função objetivo**.

# Busca Local

- **Principais Algoritmos:**
  - Hill Climbing (Busca de Subida de Encosta)
  - Simulated Annealing (Busca de Têmpera Simulada)
  - Local Beam (Busca de Feixe Local)
  - **Genetic Algorithms (Algoritmos Genéticos)**

# Relembrando as aulas de Biologia

- Todo organismo vivo é formado por uma ou mais **células**.
- Em cada célula existe um conjunto de **cromossomos**.
  - Seres humanos possuem 23 pares de cromossomos por célula.
  - O número de pares varia de espécie para espécie.
- Um cromossomo é uma longa sequência de **DNA**.
  - DNA: molécula que codifica toda a informação genética necessária para o desenvolvimento e o funcionamento dos organismos vivos.
- Um cromossomo possui vários **genes** (blocos de sequências de DNA).
- Cada gene tem uma posição própria no cromossomo.



As setas indicam uma espécie de caminho decrescente em termos do tamanho das estruturas.

Imagem modificada de: [http://en.wikipedia.org/wiki/File:Eukaryote\\_DNA-en.svg](http://en.wikipedia.org/wiki/File:Eukaryote_DNA-en.svg)

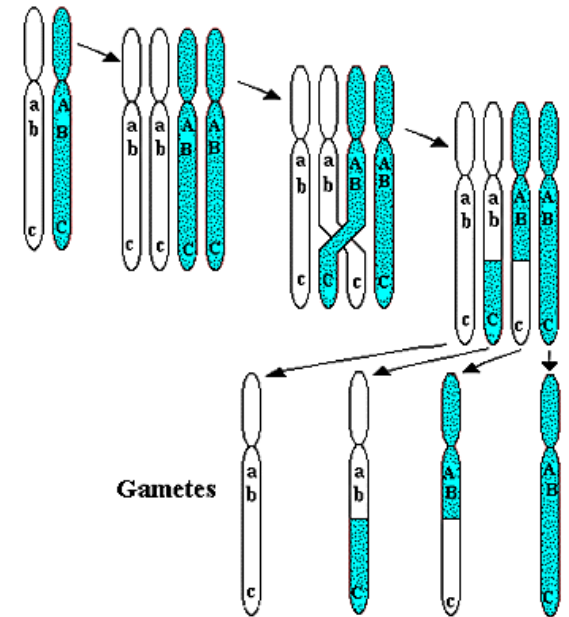
# Relembrando as aulas de Biologia

- Na natureza existem dois tipos de **reprodução**:
  - **Assexuada**: um único progenitor que se divide por mitose, como as bactérias.
  - **Sexuada**: fusão de dois organismos, na maioria das vezes de sexos opostos, que trocam material genético.
    - É a base dos **algoritmos genéticos**.



# Relembrando as aulas de Biologia

- **Reprodução sexuada:** a formação de um novo indivíduo ocorre a partir da combinação de duas células gametas.
- Na formação das células gametas, ocorre o processo de **recombinação genética** (crossing-over).
- Processo de replicação do DNA:
  - Pequenos **erros** podem ocorrer ao longo do tempo, gerando **mutações** dentro do código genético.
  - Estas mutações podem ser boas, ruins ou neutras.



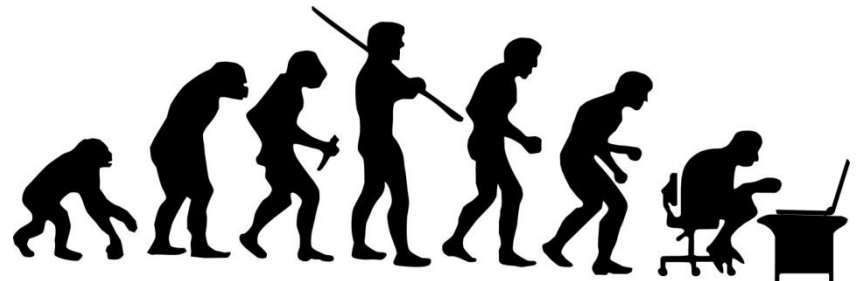
Crossing-over and recombination during meiosis

# Relembrando as aulas de Biologia

- Indivíduos que se adequam melhor ao meio ambiente (**melhor fitness**) se reproduzem mais.
  - Maior chance de passar seus genes para a **próxima geração**.
- A **recombinação** (crossover) e a **mutação** permitem que os cromossomos dos filhos não sejam exatamente iguais aos dos pais.
- Desta forma os indivíduos podem **evoluir** e se **adaptar** cada vez mais ao meio ambiente em que vivem.

# Computação Evolutiva

- Um conjunto de técnicas inspiradas na teoria da evolução de Darwin.
- Teoria da Evolução de Darwin – **Seleção Natural**
  - Os indivíduos de uma espécie competem entre si e somente os mais aptos sobrevivem e reproduzem.
  - Durante a reprodução, os novos indivíduos herdam as características dos pais, mas estão também sujeitos à mutações.
  - Espera-se que novas gerações sejam cada vez mais próximas da perfeição.





# Algoritmos Genéticos



# Algoritmos Genéticos

- Sub-área da Computação Evolutiva.
- São algoritmos matemáticos inspirados nos mecanismos de **evolução natural** e **recombinação genética**.
- Fornece um mecanismo de busca adaptativa que se baseia no princípio Darwiniano de reprodução e **sobrevivência dos mais aptos**.
- **Buscam** (dentro da atual população) as **soluções** que possuem as **melhores características** e tentam combiná-las de forma a gerar soluções ainda melhores.



# Algoritmos Genéticos

- O modelo representa uma população de indivíduos:
  - cada indivíduo é chamado de ***cromossomo***.
- A **função de aptidão (*fitness*)** avalia quais os indivíduos mais aptos em cada geração.
- Os mais aptos são mantidos e os menos aptos substituídos por novos indivíduos obtidos a partir dos mais aptos (descendentes).



# Algoritmos Genéticos: Processo

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

# Algoritmos Genéticos

Passos para criar algoritmos genéticos:

1. Codificar a **população de indivíduos**.
2. Definir uma **função de aptidão**.
3. Definir um método de **seleção dos pais**.
4. Definir os **operadores genéticos**:
  - Recombinação (crossover)
  - Mutação
5. Definir um **critério de parada**



# Algoritmos Genéticos

Passos para criar algoritmos genéticos:

1. Codificar a **população de indivíduos**.
2. Definir uma **função de aptidão**.
3. Definir um método de **seleção dos pais**.
4. Definir os **operadores genéticos**:
  - Recombinação (crossover)
  - Mutação
5. Definir um **critério de parada**



# 1. Codificar a População de Indivíduos

- Representação dos Cromossomos:
  - Traduz a informação do problema em uma forma viável de ser tratada pelo computador.
  - Uma cadeia de genes que representam cada indivíduo da população.
    - **Genes:** representação de algum parâmetro, por meio de alfabeto, valores inteiros, caracteres, valores binários, etc.
- Cromossomos podem ser formados por:
  - (0101 ... 1100) valores binários
  - (43.2 -33.1 ... 0.0 89.2) números reais
  - (3 5... 10 87 1110) números inteiros
  - (R1 R2 R3 ... R22 R23) listas de regras
  - Qualquer estrutura de dados imaginável!



# 1. Codificar a População de Indivíduos

- Um **indivíduo** corresponde a um cromossomo.
- A **população** é formada por um conjunto de indivíduos que irão competir pela sobrevivência e pela reprodução, com o objetivo de perpetuar as suas características.
- **Exemplo 1:** Problema do Caixeiro Viajante:
  - Um indivíduo pode ser:

A	G	C	B	D	F	E
---	---	---	---	---	---	---





# 1. Codificar a População de Indivíduos

## Exemplo 2:

- Se define o valor de cada gene do cromossomo para todos os indivíduos da população, indicando a presença ou ausência de determinada característica:
  - 1: presença de determinada característica
  - 0: ausência de determinada característica
- Conjunto de cromossomos (indivíduos) que compõem a população:

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

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



# 1. Codificar a População de Indivíduos

- Estratégia simples: **gerar a população inicial aleatoriamente.**
- Grande número de indivíduos aumenta a diversidade, permitindo explorar melhor as habilidades da população.
  - **Grandes populações:** exigem maior poder computacional.
  - **Pequenas populações:** pode ser necessário aumentar a taxa de mutação para permitir a exploração de um espaço maior.

# Algoritmos Genéticos

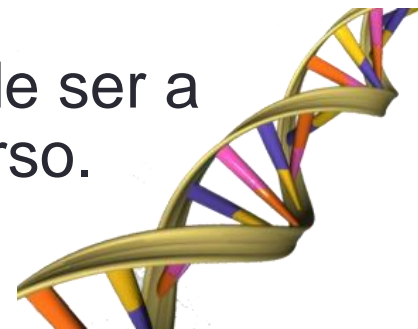
Passos para criar algoritmos genéticos:

1. Codificar a **população de indivíduos**.
2. Definir uma **função de aptidão**.
3. Definir um método de **seleção dos pais**.
4. Definir os **operadores genéticos**:
  - Recombinação (crossover)
  - Mutação
5. Definir um **critério de parada**



## 2. Definir uma Função de Aptidão (*fitness*)

- Determina a **qualidade de um indivíduo** como solução do problema em questão.
- São representadas por funções matemáticas
  - Devem ser maximizadas ou minimizadas, de acordo com o problema a ser resolvido.
- **Deve ser escolhida cuidadosamente**
  - Deve embutir todo o conhecimento que se possui sobre o problema a ser resolvido.
- **Exemplo:** no problema do caixeiro viajante pode ser a soma das distâncias entre as cidades do percurso.



# Algoritmos Genéticos

Passos para criar algoritmos genéticos:

1. Codificar a **população de indivíduos**.
2. Definir uma **função de aptidão**.
3. Definir um método de **seleção dos pais**.
4. Definir os **operadores genéticos**:
  - Recombinação (crossover)
  - Mutação
5. Definir um **critério de parada**



### 3. Definir um método de Seleção dos Pais

- Simula o mecanismo de seleção natural, onde os pais com **maior capacidade geram mais filhos**, porém os menos aptos também podem gerar descendentes.
- Se privilegia os indivíduos com **função de aptidão alta**, sem desprezar completamente aqueles indivíduos com função de aptidão extremamente baixa.
  - Até indivíduos com péssima aptidão podem ter características genéticas que sejam favoráveis à criação de um “indivíduo ideal”.



### 3. Definir um método de Seleção dos Pais

- Métodos de seleção de pais:
  - **Seleção por torneio:** 3 indivíduos são escolhidos ao acaso e o melhor deles, segundo a função de aptidão, será escolhido para a reprodução.
    - Repete até achar um individuo a altura para fazer o cruzamento.



### 3. Definir um método de Seleção dos Pais

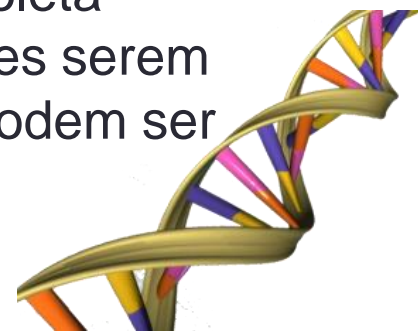
- Método de seleção de pais:
  - **Seleção por torneio:** 3 indivíduos são escolhidos ao acaso e o melhor deles, segundo a função de aptidão, será escolhido para a reprodução.
    - Repete até achar um par a altura para fazer o cruzamento.
  - **Seleção aleatória:** dois indivíduos da população são escolhidos ao acaso.





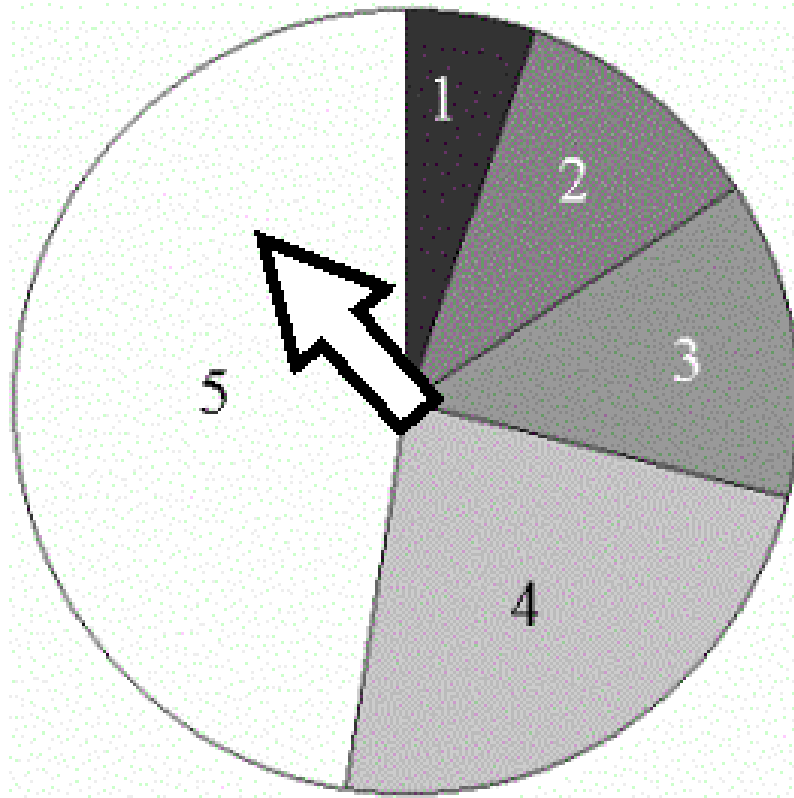
### 3. Definir um método de Seleção dos Pais

- Método de seleção de pais:
  - **Seleção por torneio:** 3 indivíduos são escolhidos ao acaso e o melhor deles, segundo a função de aptidão, será escolhido para a reprodução.
    - Repete até achar um par a altura para fazer o cruzamento.
  - **Seleção aleatória:** dois indivíduos da população são escolhidos ao acaso.
  - **Seleção usando a roleta:** as habilidades dos indivíduos são calculadas e distribuídas proporcionalmente em uma roleta (virtual). Ao girar a roleta, a chance dos setores maiores serem selecionados será maior. Porém, os demais também podem ser selecionados.



### 3. Definir um método de Seleção dos Pais

- Exemplo de Método de seleção de pais:
  - **Seleção usando a roleta:**



# Algoritmos Genéticos

Passos para criar algoritmos genéticos:

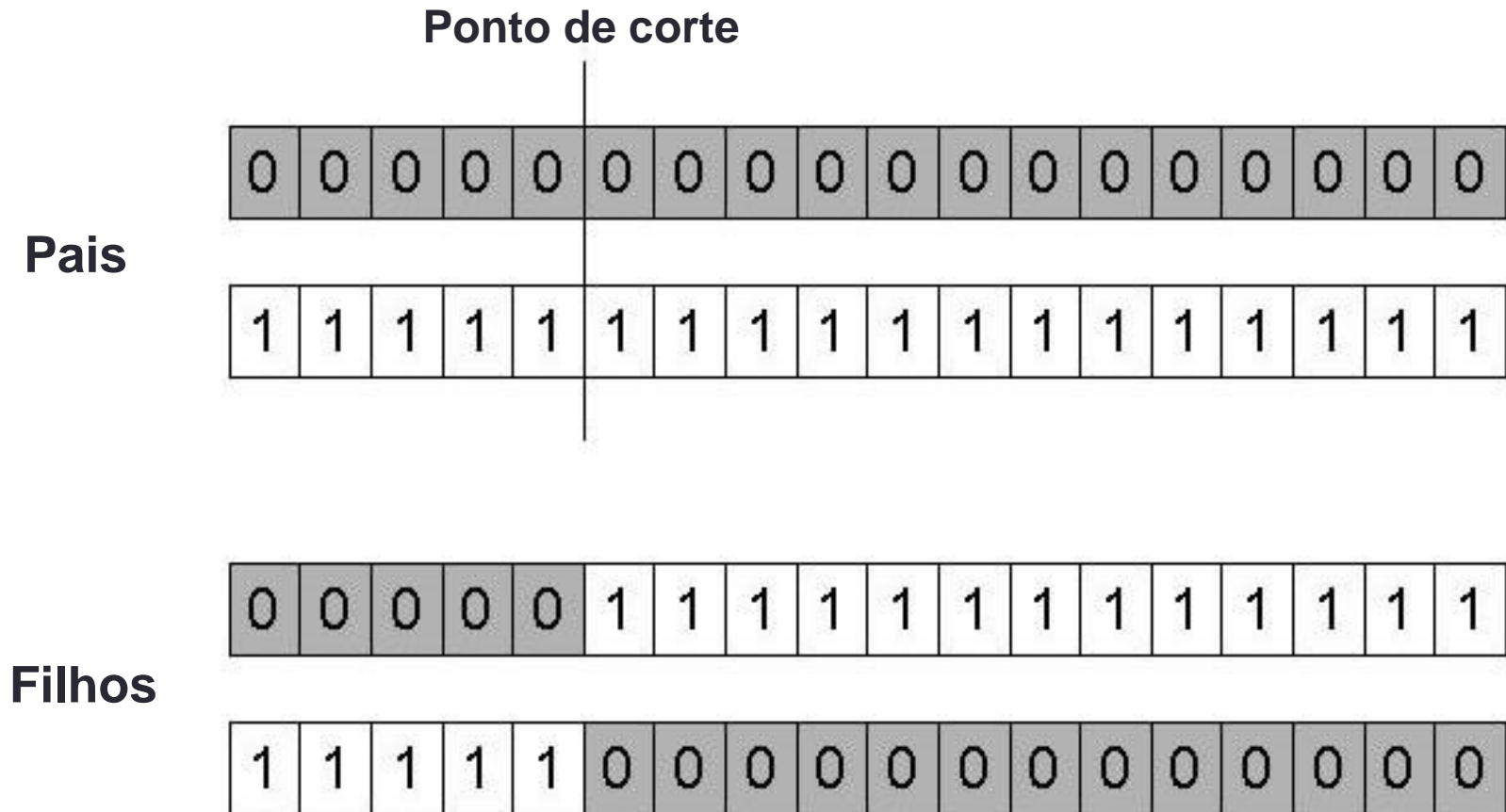
1. Codificar a **população de indivíduos**.
2. Definir uma **função de aptidão**.
3. Definir um método de **seleção dos pais**.
4. Definir os **operadores genéticos**:
  - Recombinação (crossover)
  - Mutação
5. Definir um **critério de parada**



## 4. Definir os operadores genéticos: Recombinação

- Operador de cruzamento (crossover) entre dois indivíduos.
- **Processo:**
  1. Selecionar 2 pais através processo de seleção de pais.
  2. Selecionar um ponto de corte (uma posição entre dois genes de um cromossomo).
    - É o ponto de separação entre cada um dos genes que compõem o material genético de cada pai.
  3. A metade à esquerda do ponto de corte vai para um filho e a metade à direita vai para outro.

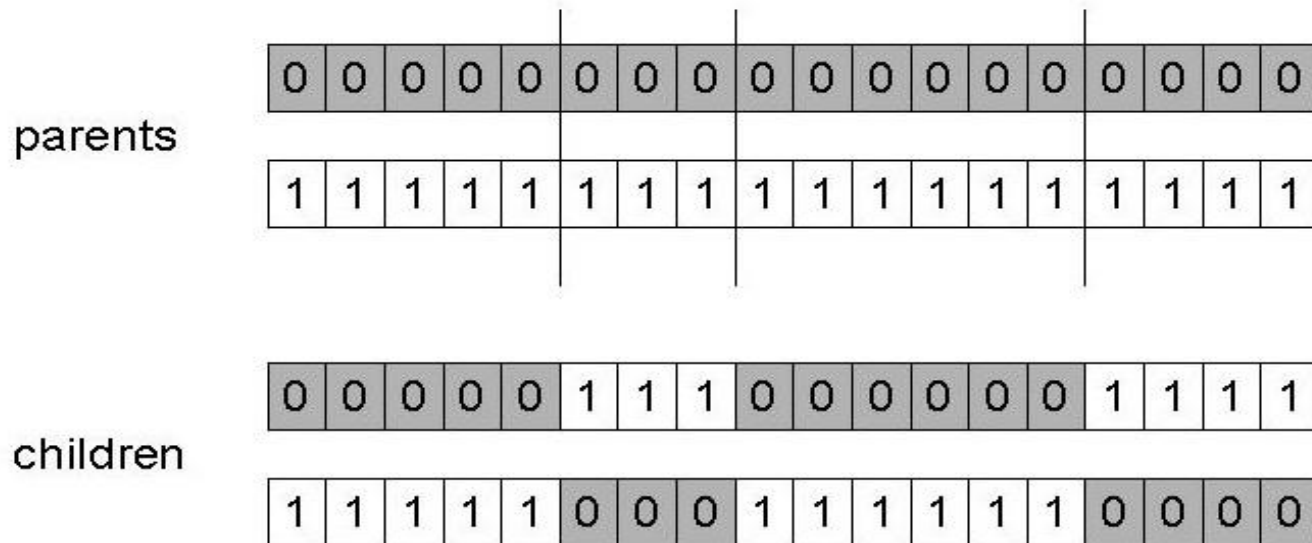
## 4. Definir os operadores genéticos: Recombinação



# Outras técnicas de Recombinação (Crossover)

## Recombinação de mais pontos

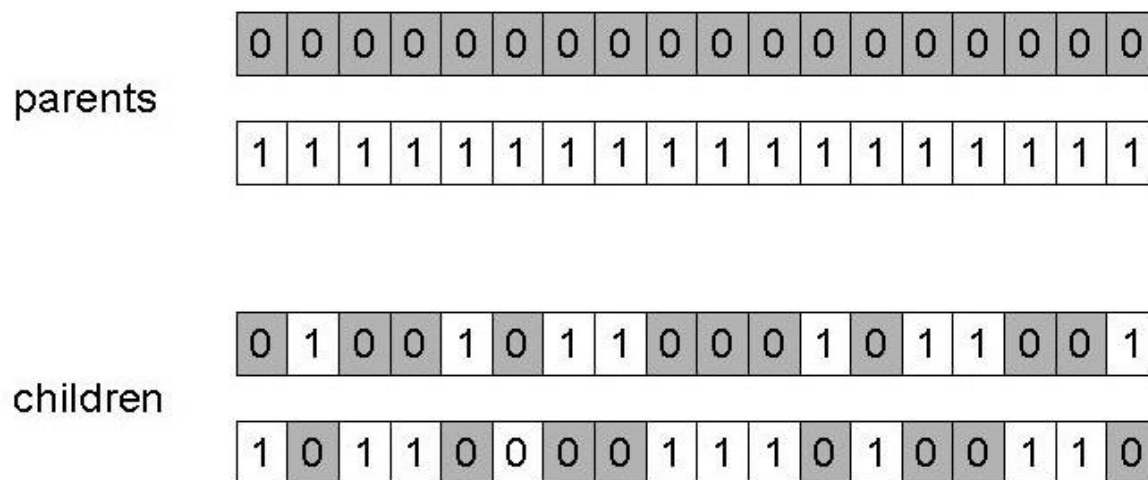
- É sorteado dois ou mais pontos de corte.
- Melhora a capacidade de geração dos conjuntos de cromossomos.



# Outras técnicas de Recombinação (Crossover)

## Recombinação uniforme

- Para cada gene é sorteado um número zero ou um.
  - Se o sorteado for 1, um filho recebe o gene do primeiro pai e o segundo filho o gene do segundo pai.
  - Se o sorteado for 0, o primeiro filho recebe o gene do segundo pai e o segundo filho recebe o gene do primeiro pai.



# Algoritmos Genéticos

Passos para criar algoritmos genéticos:

1. Codificar a **população de indivíduos**.
2. Definir uma **função de aptidão**.
3. Definir um método de **seleção dos pais**.
4. Definir os **operadores genéticos**:
  - Recombinação (crossover)
  - Mutação
5. Definir um **critério de parada**





## 4. Definir os operadores genéticos: Mutação

- Nesta etapa, o cromossomo pode ser modificado em algumas de suas partes, assumindo características que não pertencem aos pais.
- A probabilidade da mutação é extremamente baixa (em torno de 1%) de alteração aleatória do valor de um gene ou mais genes dos filhos.
- A mutação mantém a variedade genética da população.

## 4. Definir os operadores genéticos: Mutação

- No caso binário, a mutação consiste em inverter alguns bits com base em uma probabilidade;
- No caso real ou inteiro, somam-se pequenos valores aleatórios (positivos ou negativos);
- No caso de uma representação usando cadeias de caracteres, alguns caracteres pode ser alterados aleatoriamente.

**Pais**

1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

**Filhos**

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

# Algoritmos Genéticos

Passos para criar algoritmos genéticos:

1. Codificar a **população de indivíduos**.
2. Definir uma **função de aptidão**.
3. Definir um método de **seleção dos pais**.
4. Definir os **operadores genéticos**:
  - Recombinação (crossover)
  - Mutação
5. Definir um **critério de parada**



## 5. Definir um critério de parada

- **Limite de Gerações:** o algoritmo para quando um número de gerações predefinido é atingido.
- **Limite de Tempo:** o algoritmo para quando um tempo máximo predefinido foi ultrapassado.
- **Sem melhor:** Se durante um número determinado de gerações nenhum indivíduo supera o melhor atual então o algoritmo pode ser interrompido.
- **Sem mudanças:** Se durante um número determinado de gerações não há mudanças na informação contida nos genótipos o algoritmo pode parar.

# Exemplo de Aplicação Algoritmos Genéticos: Seleção de Rotas

# Exemplo de Aplicação Algoritmos Genéticos: Seleção de Rotas

- **Problema do caixeiro viajante:**
  - Encontrar o caminho mais curto para percorrer  $n$  cidades sem repetição.
- **Como representar os indivíduos?**
  - Cada indivíduo pode ser representador por uma lista ordenada de cidades, que indica a ordem em que cada uma será visitada.
  - **Exemplo:** (3 5 7 2 1 6 4 8)



# Exemplo: Algoritmos Genéticos

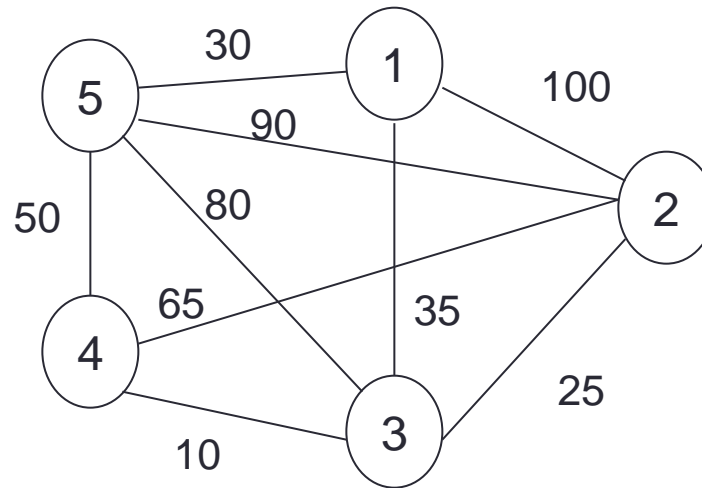
- Cada cromossomo deve conter **todas as cidades** do percurso, **apenas uma vez**.
- **Considerando 8 cidades:**
  - **Cromossomos válidos:**
    - (1 2 3 4 5 6 7 8), (8 7 6 5 4 3 2 1), (1 3 5 7 2 4 6 8)...
  - **Cromossomos inválidos:**
    - (1 5 7 8 4 3 6) - Falta a cidade 2,
    - (1 5 7 8 4 3 6 7) - Falta a cidade 2 e a cidade 7 está representada 2 vezes...



# Exemplo: Algoritmos Genéticos

- **Qual a Função de Aptidão (fitness)?**
  - Neste caso é a soma de todas as distâncias entre cidades consecutivas.

- **Exemplo:**



- O cromossomo (1 3 5 4 2) tem aptidão igual a:
  - $35 + 80 + 50 + 65 = 230$





# Exemplo: Algoritmos Genéticos

## Recombinação (uniforme):

Pai1 (3 5 7 2 1 6 4 8)

Pai2 (2 5 7 6 8 4 3 1)

1) Gerar uma string de bits aleatória do mesmo tamanho que os pais:

1 0 0 1 0 1 0 1

2) Copiar para o Filho 1 os elementos do Pai1 referentes as posições onde a string de bits possui um 1:

3 \_ \_ 2 \_ 6 \_ 8

- Elementos não copiados do Pai1:

5 7 1 4

3) Permutar essa lista de forma que os elementos apareçam na **mesma ordem** que no Pai2 e copia-los para dentro do Filho1:

3 5 7 2 4 6 1 8

# Exemplo: Algoritmos Genéticos

- **Mutação:**

- Indivíduo (3 5 7 2 4 6 1 8)
- Escolher dois elementos aleatórios dentro do cromossomo e trocar as suas posições:

(3 5 7 2 4 6 1 8)

Novo indivíduo a partir da mutação:

(3 4 7 2 5 6 1 8)



# Quando utilizar Algoritmos Genéticos?

- Para resolver aqueles problemas cujos algoritmos exatos são extremamente lentos ou incapazes de obter uma solução.
- São extremamente dependentes de **fatores estocásticos** (probabilísticos), tanto na fase de **inicialização** da população quanto na fase de **evolução**.
  - Isto faz com que os seus **resultados raramente** sejam perfeitamente reprodutíveis.
- Os algoritmos genéticos são **heurísticas** que não garantem a obtenção do melhor resultado possível em todas as suas execuções.

# Definição de um problema em algoritmos genéticos

- Representação dos indivíduos.
- Parâmetros do sistema (tamanho da população, taxa de mutação...).
- Políticas de seleção e eliminação de indivíduos.
- Operadores genéticos (recombinação e mutação)
- Critérios de parada.
- Função de avaliação (a mais importante e mais complicada de ser definida).