

# Inteligência Artificial

## Aula 09 – Algoritmos Genéticos

Msc Gustavo Molina

<msc.gustavo.unip@gmail.com>

# Métodos de Busca

- **Busca Cega ou Exaustiva:**

- Não sabe qual o melhor nó da fronteira a ser expandido. Apenas distingue o estado objetivo dos não objetivos.


- **Busca Heurística:**

- Estima qual o melhor nó da fronteira a ser expandido com base em funções heurísticas.

- **Busca Local:**

- Operam em um único estado e movem-se para a vizinhança deste estado.

# Algoritmos Genéticos

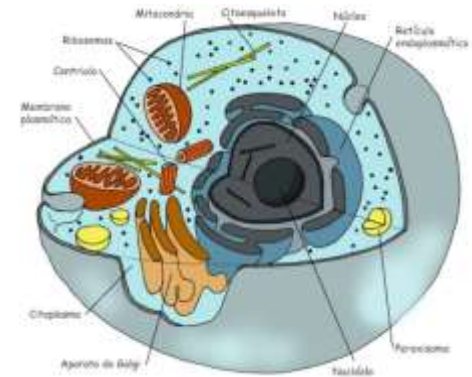
- Método de **busca local**.
  - Boa abordagem para lidar com **espaços de busca muito grandes**.
    - Possibilita navegá-los procurando por soluções que talvez não fossem encontradas em uma busca convencional mesmo que ela durasse centenas de anos.
  - Baseado na **evolução dos seres biológicos**.
- 

# Teoria da Evolução

- A **teoria da evolução** diz que na natureza todos os indivíduos dentro de um ecossistema **competem** entre si por recursos limitados (comida, água...)
- Os **indivíduos mais fracos** de uma mesma espécie tendem a não se proliferarem.
- A descendência reduzida faz com que a probabilidade de ter seus **genes propagados** ao longo de sucessivas gerações seja menor.
- A combinação entre os genes dos indivíduos que sobrevivem pode produzir um novo **indivíduo muito melhor adaptado** às características de seu meio ambiente ao combinar características possivelmente positivas de cada um dos seus pais.

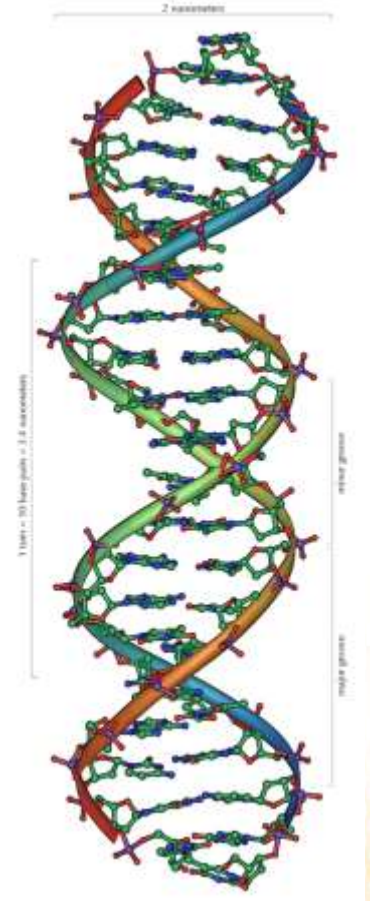
# Relembrando as Aulas de Biologia

- Todo indivíduo biológico é formado por uma ou mais **células**.
- Dentro de cada célula existe um conjunto de **cromossomos**.
- Os seres humanos têm 23 pares de cromossomos por célula.
  - O número de pares varia de espécie para espécie.



# Relembrando as Aulas de Biologia

- Um cromossomo consiste em sequências de **DNA**.
  - DNA = molécula que codifica toda a informação necessária para o desenvolvimento e funcionamento de organismos vivos.
- Um cromossomo possui vários **genes** (blocos de sequências de DNA).
- Cada gene tem uma posição própria no cromossomo.
- O conjunto completo de material genético (todos os cromossomos) é chamado de **genoma**.
- A **qualidade de um indivíduo** (fitness) é medida pelo seu sucesso (sobrevivência)

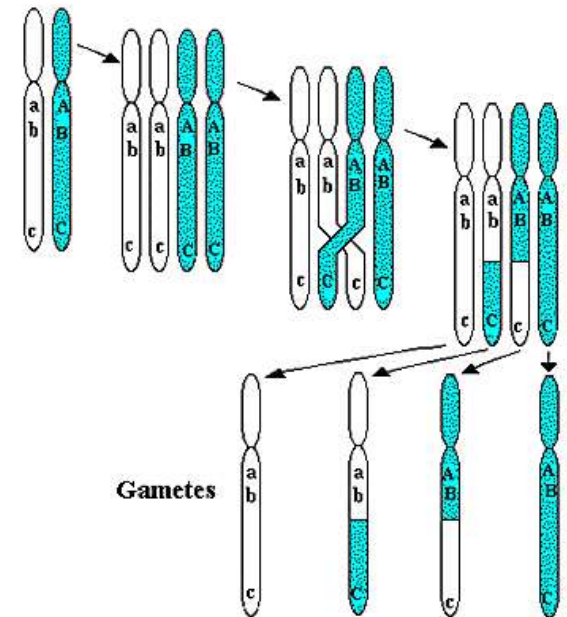


# Relembrando as Aulas de Biologia

- Na natureza existem dois tipos de **reprodução**:
  - **Assexuada**: típica de organismos inferiores, como bactérias.
  - **Sexuada**: exige a presença de dois organismos, na maioria das vezes de sexos opostos, que trocam material genético.
- **Reprodução assexuada** é base para o algoritmo de busca local Beam Search.
- **Reprodução sexuada** é a base dos **algoritmos genéticos**.

# Relembrando as Aulas de Biologia

- Na **reprodução sexuada** ocorre a formação de um **novo indivíduo** através da combinação de duas células gametas.
- Na formação destas gametas, ocorre o processo de **recombinação genética** (crossing-over).

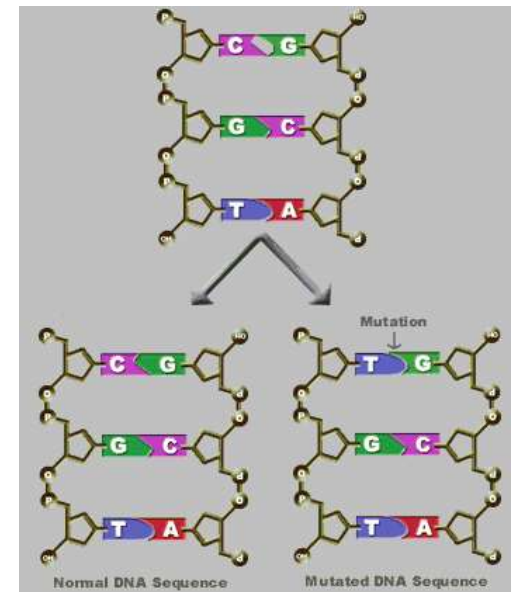


Crossing-over and recombination during meiosis



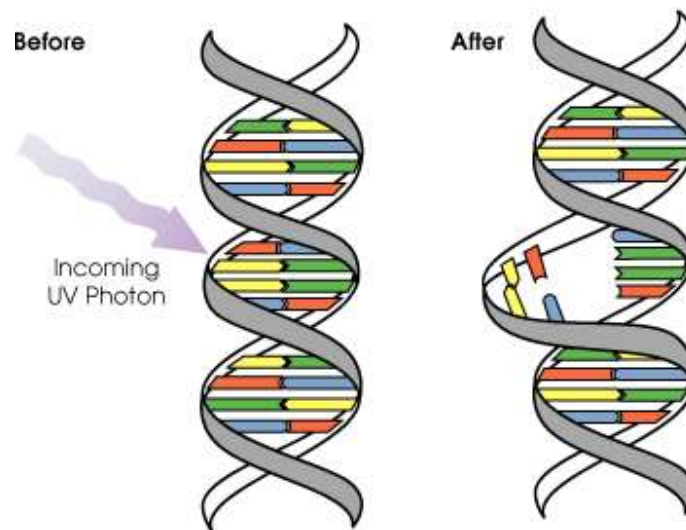
# Relembrando as Aulas de Biologia

- O processo de replicação do DNA é extremamente complexo.
- Pequenos **erros** podem ocorrer ao longo do tempo, gerando **mutações** dentro do código genético.
- Estas mutações podem ser boas, ruins ou neutras.



# Relembrando as Aulas de Biologia

- Alguns **fatores externos**, como a radiação ultravioleta, também podem causar pequenas **disrupções** no código genético.



# Relembrando as Aulas de Biologia

- Indivíduos com uma melhor adequação do seu fenótipo ao meio ambiente (**melhor fitness**) se reproduzem mais.
- Dessa forma têm mais chances de passar seus genes para a **próxima geração**.
- Entretanto, graças aos **operadores genéticos** (recombinação e mutação) os cromossomos dos filhos não são exatamente iguais aos dos pais.
- Assim, eles podem **evoluir** e se **adaptar** cada vez mais aos meio ambiente que os cerca.


# Algoritmos Evolucionários

- Os **algoritmos evolucionários**, dos quais os algoritmos genéticos fazem parte, procuram se inspirar na forma como a natureza funciona.
  - Algoritmos Genéticos
  - Programação Genética
  - Neuro-Evolução
  - Evolução Diferencial
- Os algoritmos evolucionários funcionam mantendo uma **população de estruturas** que **evoluem** de forma semelhante à evolução das espécies.

# Algoritmos Evolucionários

- Nestas estruturas são aplicados **operadores genéticos**, como a **recombinação** e **mutação**.
- Cada indivíduo recebe uma **avaliação (fitness)** que é uma quantificação da sua **qualidade** como solução do problema em questão
- Baseados nesta avaliação são aplicados operadores genéticos de forma a simular a **sobrevivência do mais apto**.

# Algoritmos Evolucionários

- Algoritmos evolucionários **buscam** (dentro da atual população) aquelas **soluções** que possuem as **melhores características** e tenta combiná-las de forma a gerar soluções ainda melhores.
  - O processo é repetido até que tenha se passado tempo suficiente ou que tenhamos obtido uma solução satisfatória para nosso problema.
- 

# Algoritmos Evolucionários

- Algoritmos evolucionários são extremamente dependente 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**.
- Além disso, claramente os algoritmos evolucionários são **heurísticas** que não garantem a obtenção do melhor resultado possível em todas as suas execuções.

# Algoritmos Evolucionários

- **Conclusão:** se você tem um algoritmo com tempo de execução razoável para solução de um problema, então não há nenhuma necessidade de se usar um algoritmo evolucionário.
- **Sempre dê prioridade aos algoritmos exatos.**
- Os algoritmos evolucionários entram em cena para resolver aqueles problemas cujos algoritmos exatos são extremamente lentos ou incapazes de obter uma solução.



# Algoritmos Genéticos

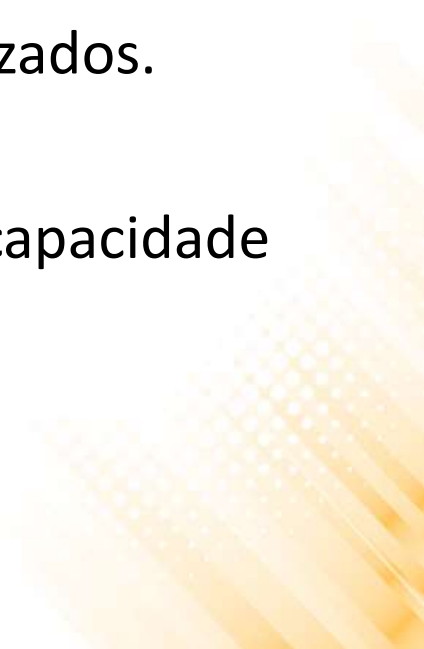
- **Algoritmos Genéticos** são uma sub-área dos Algoritmos Evolucionários. Logo, são uma metáfora para a evolução natural.



# Algoritmos Genéticos

- Nos algoritmos genéticos as populações de indivíduos são criadas e submetidas a operadores genéticos.
  - Seleção.
  - Recombinação.
  - Mutação.
- Estes operadores utilizam uma caracterização da **qualidade de cada indivíduo** como solução do problema em questão chamada de avaliação do indivíduo (**fitness**).
- É gerado um processo de **evolução natural** destes indivíduos.

# Algoritmos Genéticos

- **Definição de um problema** em algoritmos genéticos:
    - É necessário definir uma maneira de codificar os **indivíduos**.
    - Definir os **operadores genéticos** que serão utilizados.
    - Definir uma **função de avaliação** para medir a capacidade de sobrevivência de cada indivíduo.
- 

# 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

- Para criar um algoritmo genéticos é necessário:
    - Definir uma maneira de codificar a **população de indivíduos**.
    - Definir uma **função de avaliação**.
    - Definir um método de **seleção dos pais**.
    - Definir os **operadores genéticos**:
      - Recombinação.
      - Mutação.
- 

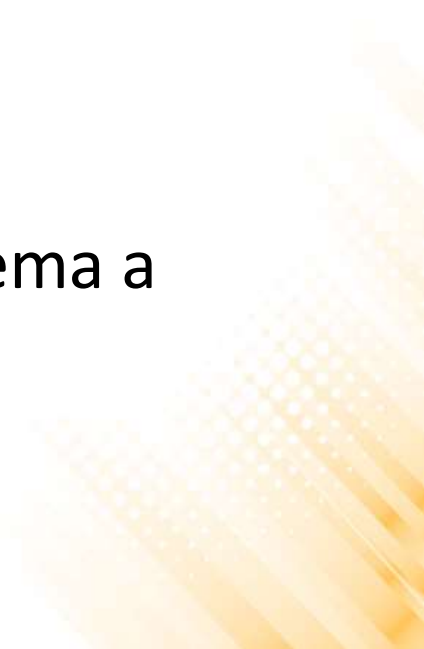
# Codificação da População

- A **representação dos cromossomos** é fundamental para o codificação do algoritmo genético.
- Consiste em uma maneira de traduzir a informação do problema em uma maneira viável de ser tratada pelo computador.
- Cada pedaço indivisível desta representação é chamado de um **gene**, por analogia aos genes que compõem um cromossomo biológico.

# Codificação da População

- É importante notar que a representação computacional dos cromossomos é **completamente arbitrária**.
- **Cromossomos podem ser:**
  - Strings de bits (0101 ... 1100)
  - Números reais (43.2 -33.1 ... 0.0 89.2)
  - Listas de regras (R1 R2 R3 ... R22 R23)
  - Qualquer estrutura de dados imaginável!

# Função de Avaliação

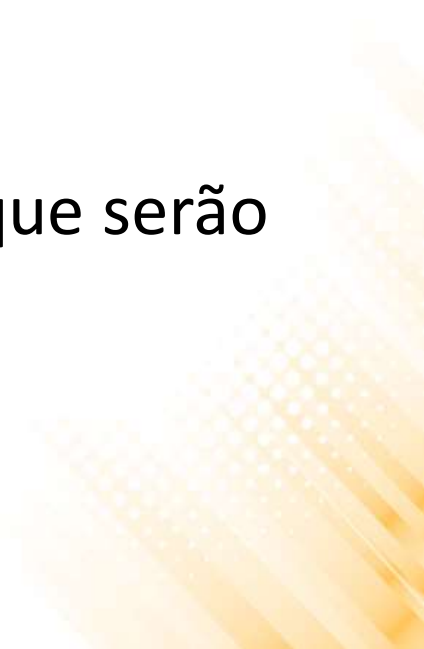
- A **função de avaliação** é a maneira utilizada pelos algoritmos genéticos para determinar a **qualidade de um indivíduo** como solução do problema em questão.
  - A **função de avaliação deve ser escolhida cuidadosamente**. Ela deve embutir todo o conhecimento que se possui sobre o problema a ser resolvido.
- 



# Seleção dos Pais

- O método de **seleção de pais** deve tentar simular o **mecanismo de seleção natural** que atua sobre as espécies biológicas.
  - Os pais mais capazes geram mais filhos, mas os menos aptos também podem gerar descendentes.
- Temos que privilegiar os indivíduos com função de avaliação alta, sem desprezar completamente aqueles indivíduos com função de avaliação extremamente baixa.
- Isto ocorre pois até indivíduos com péssima avaliação podem ter características genéticas que sejam favoráveis à criação de um "**super indivíduo**".

# Seleção dos Pais

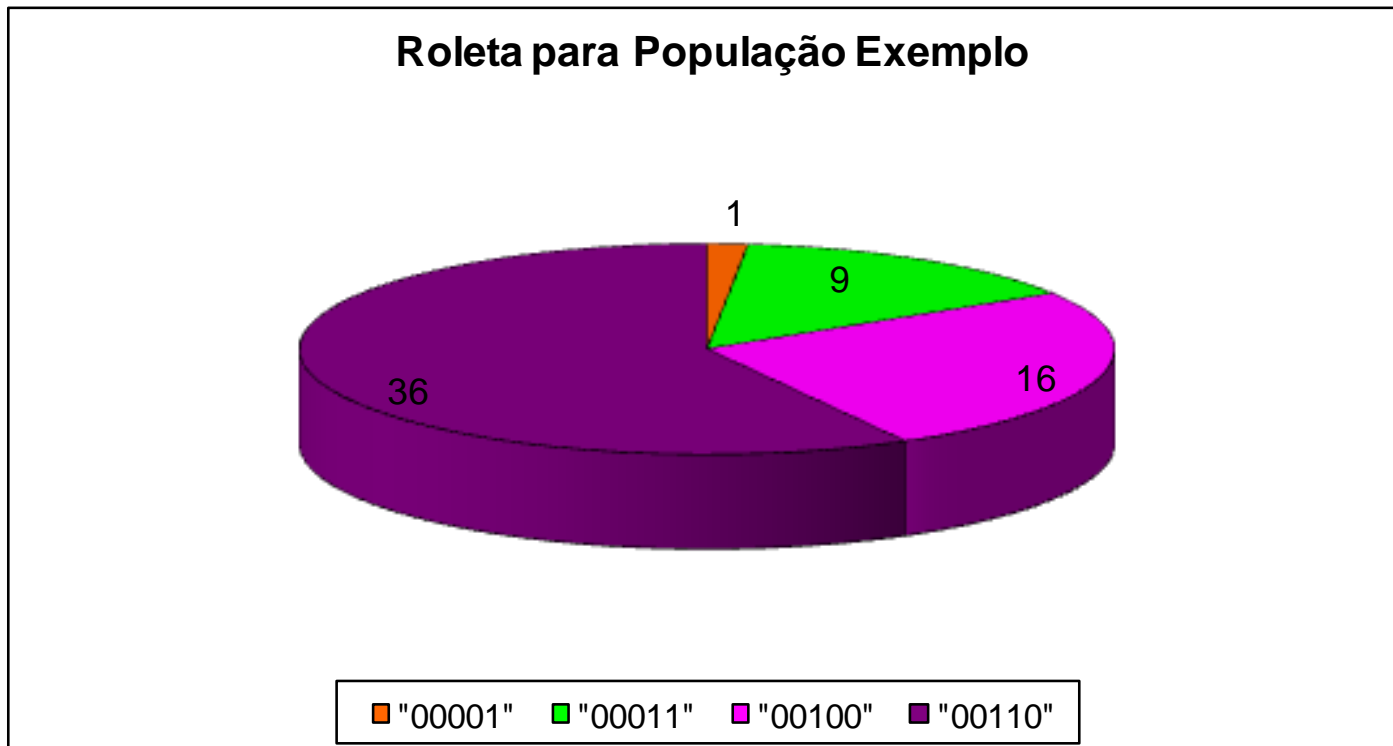
- Método mais comum de seleção de pais: **Roleta**.
  - Cria-se uma roleta (virtual) na qual cada cromossomo recebe um pedaço proporcional à sua avaliação.
  - Roda-se a roleta para sortear os indivíduo que serão pais de um novo indivíduo.
- 

# Exemplo - Seleção dos Pais

- Considerando a seguinte população gerada aleatoriamente para o problema de maximização de  $f(x)=x^2$  no intervalo  $[0,31]$

Indivíduo	Avaliação	Pedaço da roleta (%)	Pedaço da roleta (°)
00001	1	1.61	5.8
00011	9	14.51	52.2
00100	16	25.81	92.9
00110	36	58.07	209.1
<b>Total</b>	62	100.00	360.0

# Exemplo - Seleção dos Pais

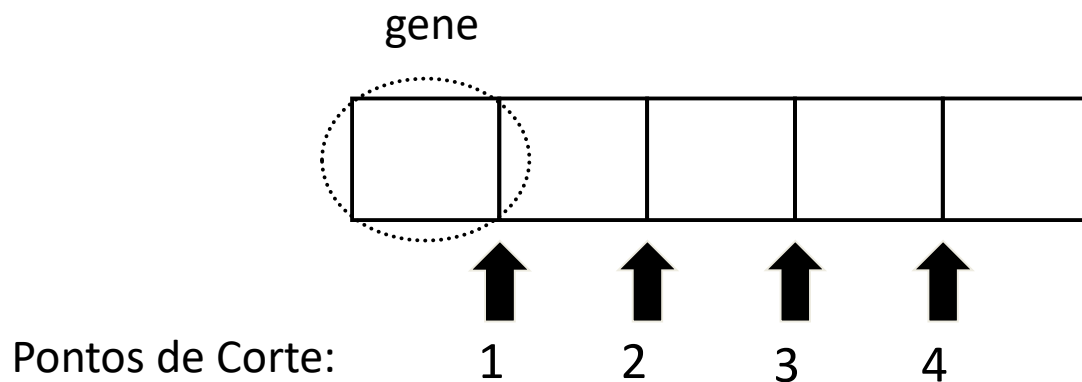


# Operadores Genéticos - Recombinação

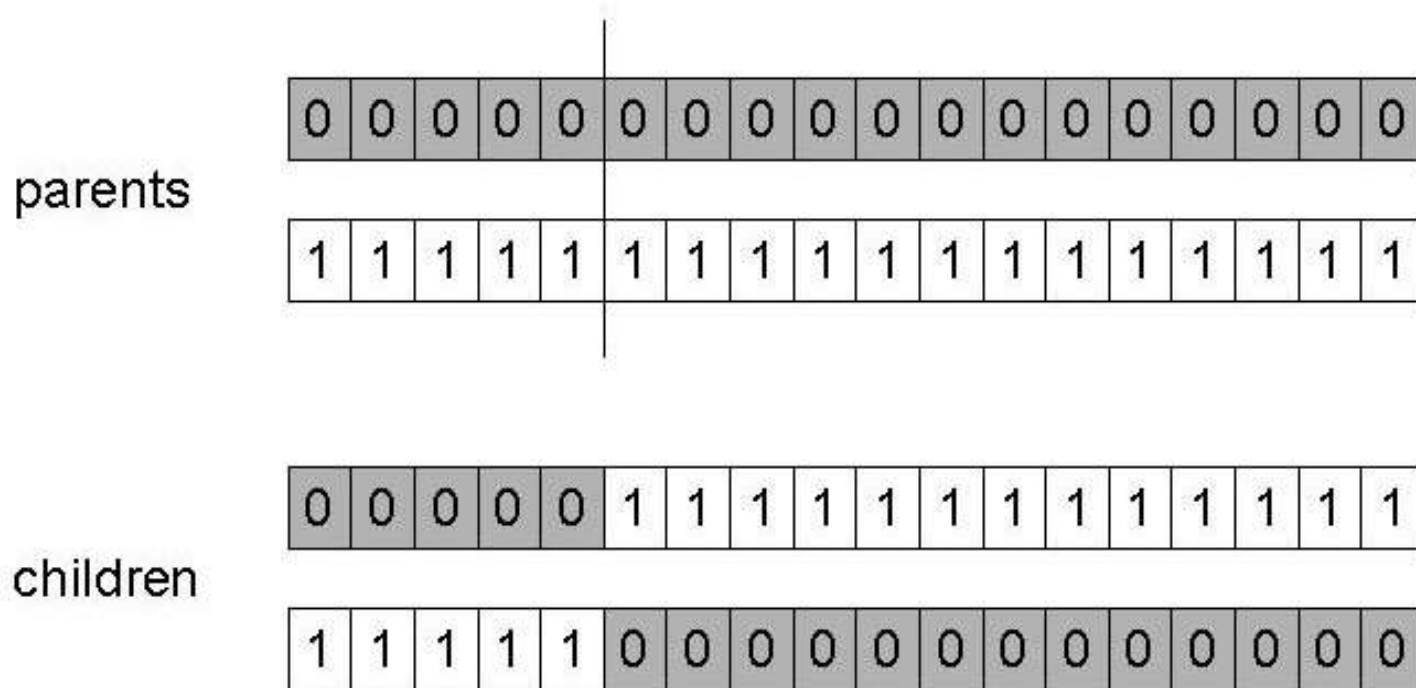
- **Operador de recombinação de um ponto.**
- **Processo:**
  - **(1)** Seleciona-se **dois pais** através processo de seleção de pais.
  - **(2)** Seleciona-se **um ponto de corte** (uma posição entre dois genes de um cromossomo). Este ponto de corte é o ponto de separação entre cada um dos genes que 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.

# Recombinação - Ponto de Corte

- Cada indivíduo com  $n$  genes possui  $n-1$  pontos de corte.
- Em um indivíduo com codificação binária, cada bit é um gene.



# Exemplo - Recombinação



# Operadores Genéticos - Mutação

- Depois de compostos os filhos, entra em ação o operador de **mutação**.
- O operador atua com base em uma **probabilidade** extremamente baixa (da ordem de 5%) de **alteração aleatória** do valor de um gene ou mais genes dos filhos.
- O valor da probabilidade que decide se o operador de mutação será ou não aplicado é um dos parâmetros do algoritmo genético que pode alterar o resultado alcançado pelo algoritmo.

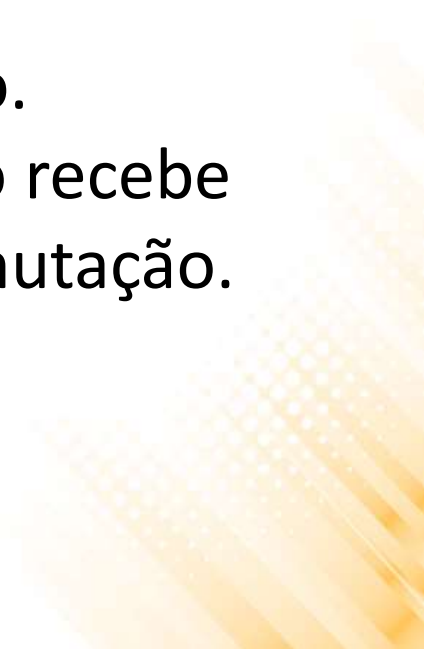


# Exemplo – Mutação

- Altere-se cada gene de forma independente com base em uma probabilidade  $p_m$
- $p_m$  é denominada taxa de mutação e costuma ser bem baixa.

parent	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	
child	0	1	0	0	1	0	1	1	0	0	0	1	0	1	1	0	0	1

# Operadores Genéticos

- É possível aumentar ou diminuir a **incidência de cada um dos operadores** sobre a população e assim ter mais controle sobre o desenvolvimento dos cromossomos.
  - Cada operador pode receber uma avaliação. Normalmente o operador de recombinação recebe um fitness bem maior que o operador de mutação.
- 

# Operadores Genéticos

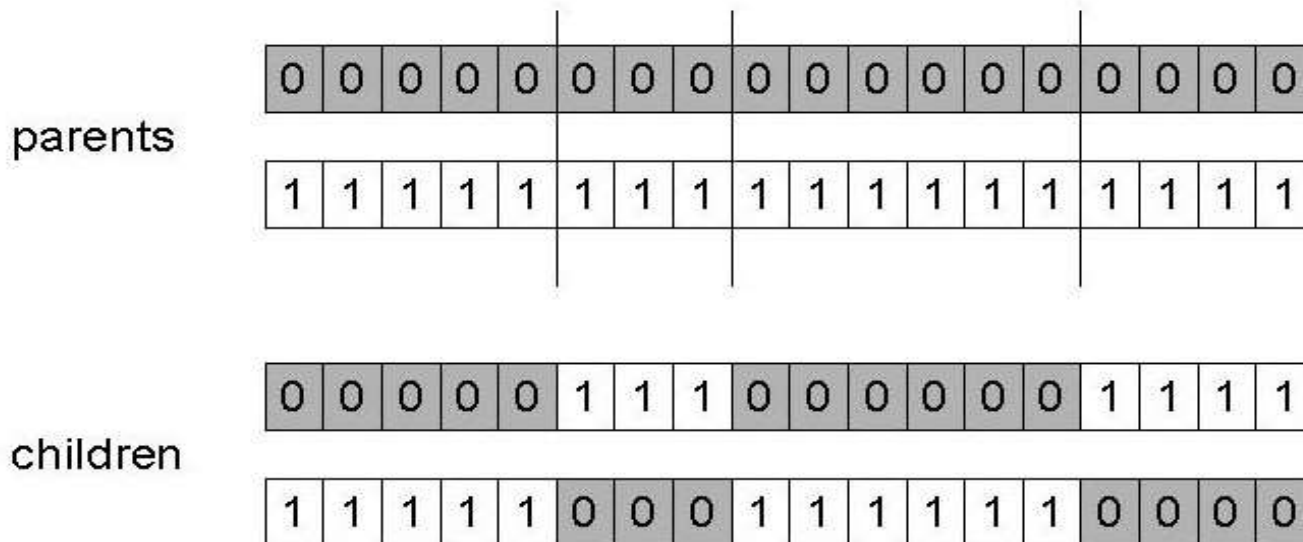
- As porcentagem de aplicação de cada operador não precisa ser fixa.
- No **início** queremos executar **muita reprodução e pouca mutação**, visto que há muita diversidade genética e queremos explorar o máximo possível nosso espaço de soluções.
- Depois de um grande número de gerações, há pouca diversidade genética na população e seria extremamente interessante que o operador de mutação fosse escolhido mais frequentemente.

# Recombinação de Dois Pontos

- Existem indivíduos que não podem ser gerados com a recombinação de somente um ponto. Exemplo: 1\*\*\*\*\*1.
- Consequentemente, se não mudarmos o operador de recombinação, o algoritmo genético fica limitado na sua capacidade de gerar um certo conjunto de cromossomos.
- Para melhorar essa capacidade é possível introduzir a recombinação de 2 pontos.
- Nele, em vez de sortearmos um só ponto de corte, sorteamos dois.

# Recombinação de n Pontos

- Evoluindo a idéia da recombinação de dois pontos, é possível tornar o operador uma recombinação de n pontos.



# 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.

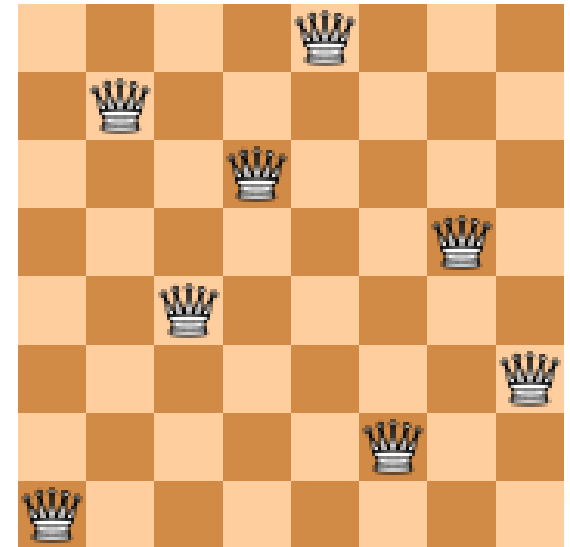
parents	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	
children	0	1	0	0	1	0	1	1	0	0	0	1	0	1	1	0	0	1
	1	0	1	1	0	0	0	0	1	1	1	0	1	0	0	1	1	0

# Elitismo

- A idéia básica por trás do elitismo é a seguinte:
  - Os  $n$  melhores indivíduos de cada geração não devem "morrer" junto com a sua geração, mas sim passar para a próxima geração para garantir que seus genomas sejam preservado.
- **É uma forma de garantir que o algoritmo nunca regrida.**

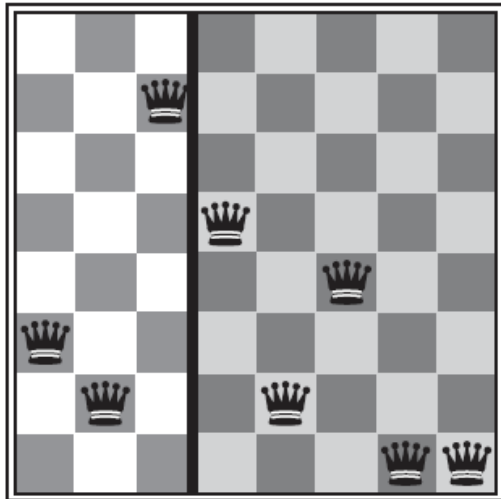
# Algoritmos Genéticos - Exemplo 1

- **Problema: 8 Rainhas**
- **Como representar os indivíduos?**
  - 8 dígitos – cada um representado a posição da rainha em sua coluna.
  - **Exemplo:** (1, 7, 4, 6, 8, 2, 5, 3)
- **Qual a função de avaliação?**
  - Número de pares de rainhas não sedo atacadas.

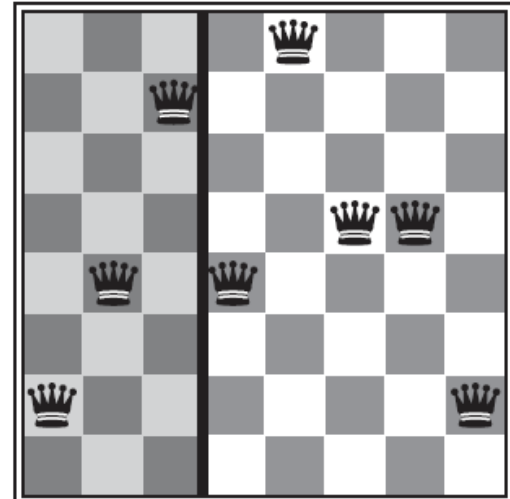




# Algoritmos Genéticos - Exemplo 1

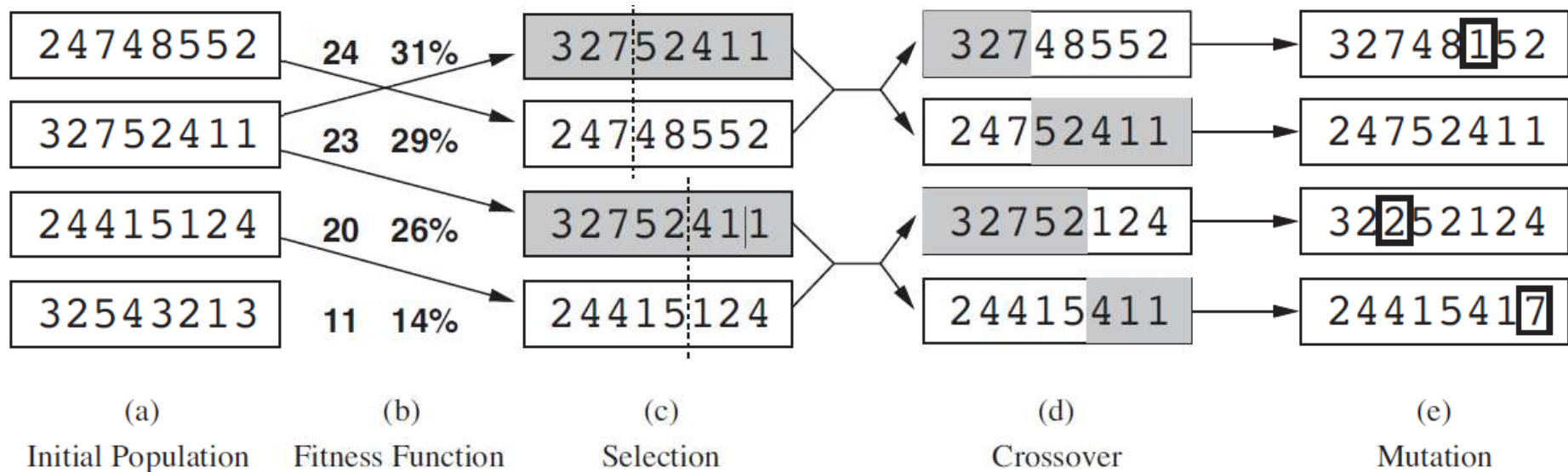


$$(3, 2, 7, 5, 2, 4, 1, 1) = 23$$



$$(2, 4, 7, 4, 8, 5, 5, 2) = 24$$

# Algoritmos Genéticos - Exemplo 1



# Algoritmos Genéticos - Exemplo 2

- **Problema do caixeiro viajante:** Deve-se 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)

# Algoritmos Genéticos – Exemplo 2

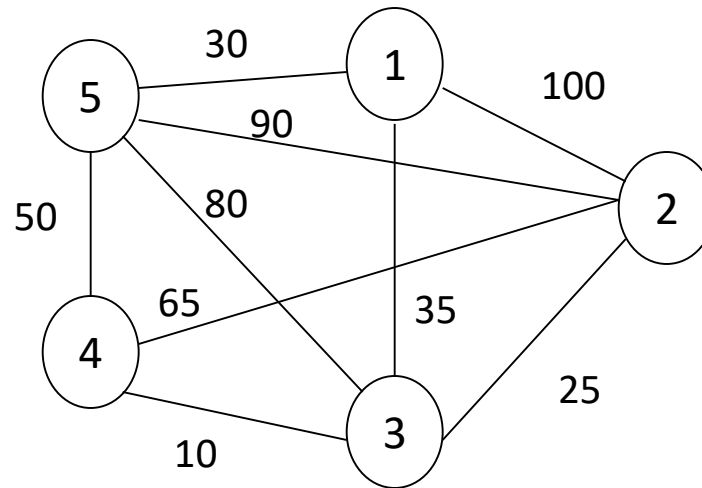
- Cada cromossomo tem que conter **todas as cidades** do percurso, **apenas uma vez**.
- **Considerando 8 cidades:**
  - **Cromossomos válidos:** (1 2 3 4 5 6 7 8), (8 7 6 5 4 3 2 1), (1 3 5 7 2 4 6 8)...
  - **Cromossomos inválidos:** (1 5 7 8 2 3 6) - Falta a cidade 4, (1 5 7 8 2 3 6 5) - Falta a cidade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

# Algoritmos Genéticos – Exemplo 2

- Qual a função de avaliação?

- A **função de avaliação** consiste em somar todas as distâncias entre cidades consecutivas.

- **Exemplo:**



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

# Algoritmos Genéticos – Exemplo 2

- **Recombinação (uniforme):**

- Pai1 (3 5 7 2 1 6 4 8)

- Pai2 (2 5 7 6 8 4 3 1)

- 1) Gera-se uma string de bits aleatória do mesmo tamanho que os pais:

1 0 0 1 0 1 0 1

- 2) Copia-se para o filho 1 os elementos do pai 1 referentes àquelas posições onde a string de bits possui um 1:

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

- 3) Elementos não copiados do pai1:

5 7 1 4

- 3) Permuta-se essa lista de forma que os elementos apareçam na mesma ordem que no pai 2 e copia-se eles para dentro do Filho1:

3 5 7 2 4 6 1 8

# Algoritmos Genéticos – Exemplo 2

- **Mutação:**

- Indivíduo (3 5 7 2 1 6 4 8)


- Escolhem-se dois elementos aleatórios dentro do cromossomo e trocam-se as suas posições:

(3 5 7 2 1 6 4 8)

Novo indivíduo mutante:

(3 1 7 2 5 6 4 8)

# Algoritmos Genéticos

- Questões importantes na 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).
- 



# Leitura Complementar

- Russell, S. and Norvig, P. **Artificial Intelligence: a Modern Approach**, 3rd Edition, Prentice-Hall, 2009.
- **Capítulo 4: Informed Search and Exploration**

