

Computação Evolucionária

Prof. Me. Alexandre Henrick

Sistemas de Informação - 8º P

Computação Evolucionária

- Linha de pesquisa da I.A que propõe um paradigma para desenvolvimento de soluções baseando-se na evolução por seleção natural (Darwin, 1859).
- Assim como outras áreas da IA, se inspira na natureza para construção de algoritmos.

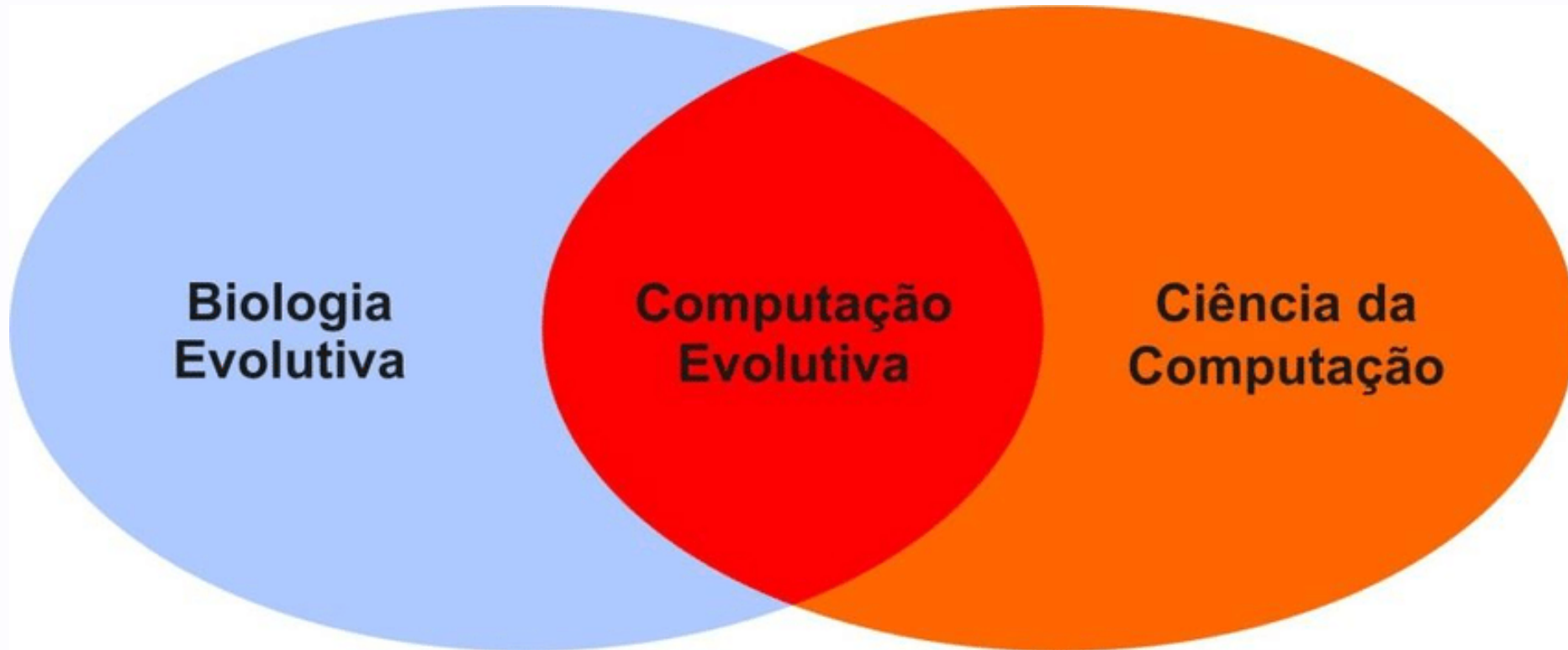


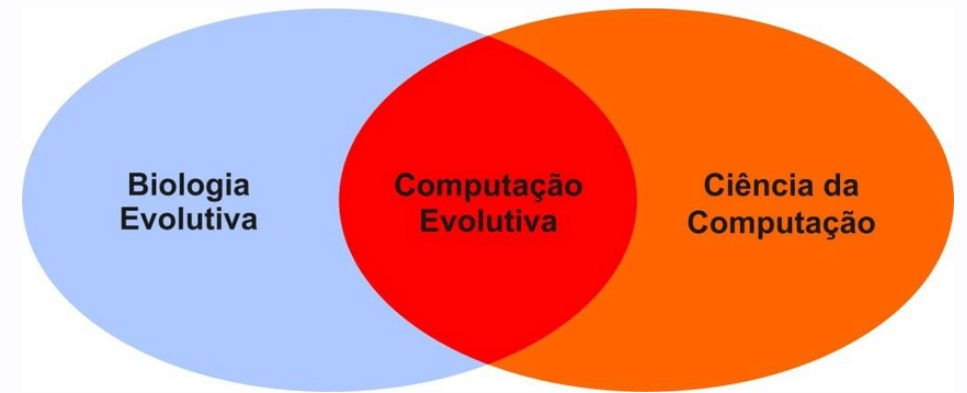
Diagrama de Venn - Computação Evolucionária

Pequeno Histórico da Computação Evolucionária

- Métodos inspirados na evolução de Darwin começam a ser propostos em 1958
- Anos 60: Algoritmos Genéticos
- 1992: Programação Genética
- Anos 90 essas técnicas foram combinadas formando a área de Computação Evolucionária

Onde usamos CE?

- Engenharia;
- Design de circuitos;
- Modelos financeiros;
- Jogos;
- Bioinformática;
- Modelagem

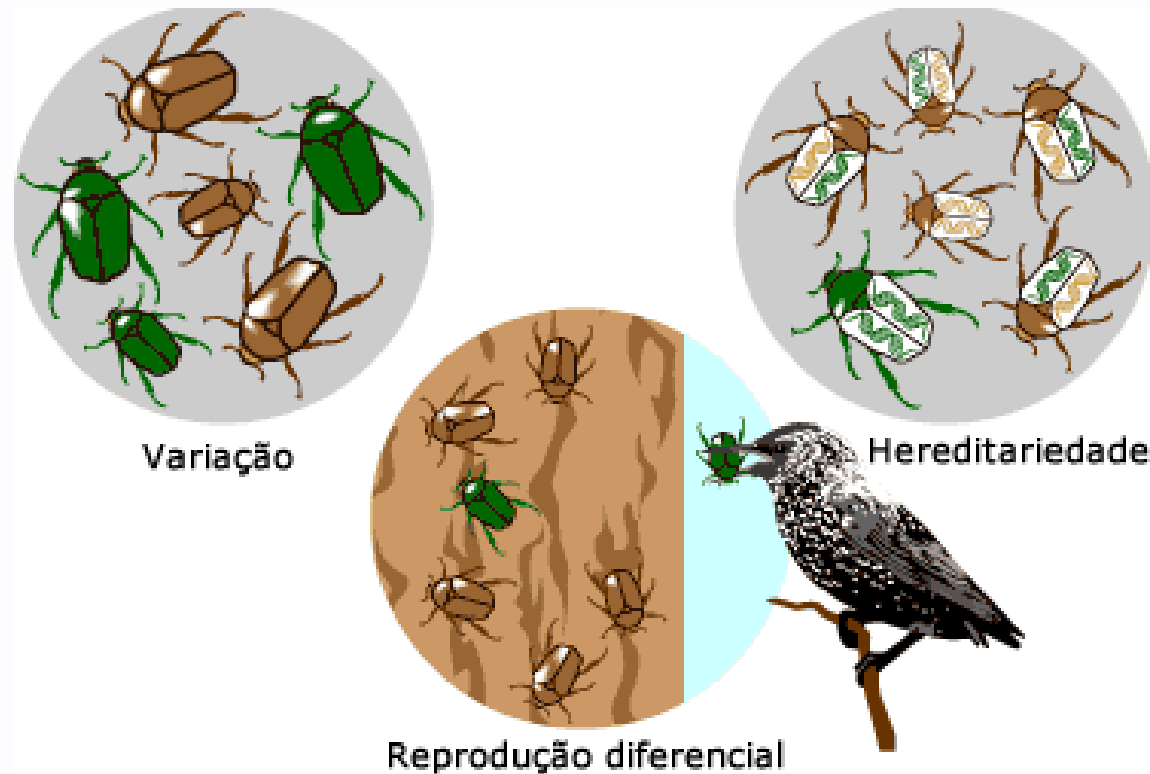


Seleção Natural

- No cenário apresentado, dizemos que os besouros morrons são mais "aptos"



As 3 características da Seleção Natural



Variabilidade Genética

- Introduz novas características que podem, ou não, serem vantajosas no ambiente que o indivíduo vive
- Aumenta a possibilidade de indivíduos mais "aptos"
- **Mutação** é um dos principais mecanismos

Reprodução Diferencial

- Os indivíduos menos aptos (besouros verdes) reproduzem menos
- Besouros marrons, por serem a maioria devido a sua aptidão, passam a reproduzir mais

Hereditariedade

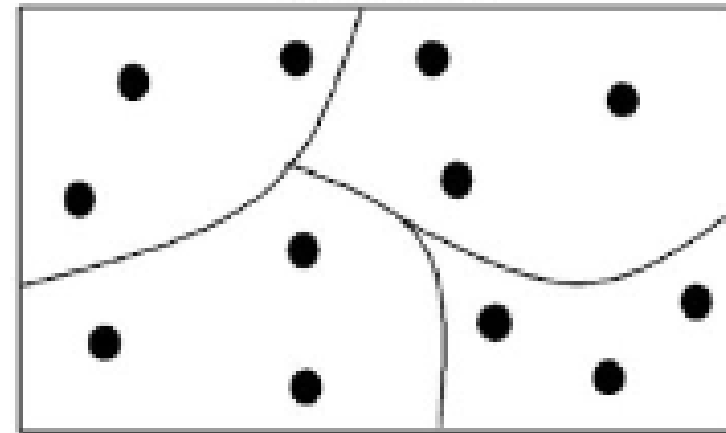
- Os besouros marrons passam suas características para gerações futuras
- Espécies mais aptas passam a existir
- A principal característica, a cor marrom, passa a ser bastante presente nas próximas gerações

Representação visual da Evolução

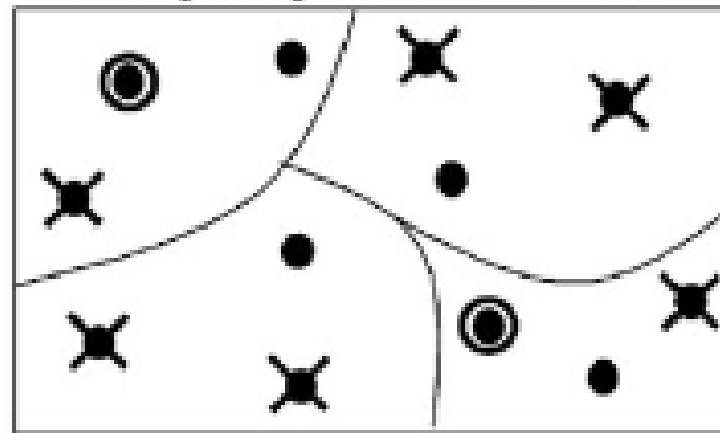
visão geral do espaço de busca



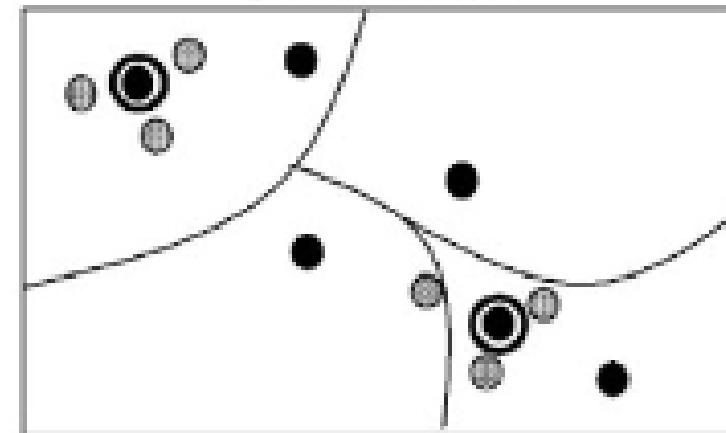
geração atual



escolha dos indivíduos que irão se reproduzir e aqueles que irão ser substituídos



próxima geração



A lacuna na teoria de Darwin

- Teoria não explicou **como** as mutações ocorriam e eram transmitidas. Onde eram "armazenadas"?
- Gregor Mendel (1865), conseguiu prever a probabilidade de um "elemento celular" ser preservado em gerações futuras

Genética Mendeliana

- Esse elemento é o que hoje chamamos de **Gene**, a unidade de hereditariedade do DNA (ácido desoxirribonucleico)
- Ou seja, essa unidade contribui para a formação em nível celular das características de qualquer espécie

Algoritmos Genéticos...

- John Henry Holland
1929 - 2015
- Através de
experiências com
RNAs, percebeu elo
entre Biologia e
Computação



- Criou modelo matemático do processo evolutivo
- **Livro: Adaptation in Natural and Artificial Systems - 1975**



- David E. Goldberg
1953 -
- Orientado por John
Henry Holland.
Estendeu seus
trabalhos sobre AGs
- **Livro: Genetic
Algorithms in
Search, Optimization
and Machine
Learning - 1989**



Algoritmos Genéticos

- Algoritmos de busca **não determinista** inspirados na Evolução
- **Busca e otimização**
- Ao longo dos anos aplicado em diferentes tipos de problemas. Ex: classificação (Machine Learning)

Algoritmos Genéticos

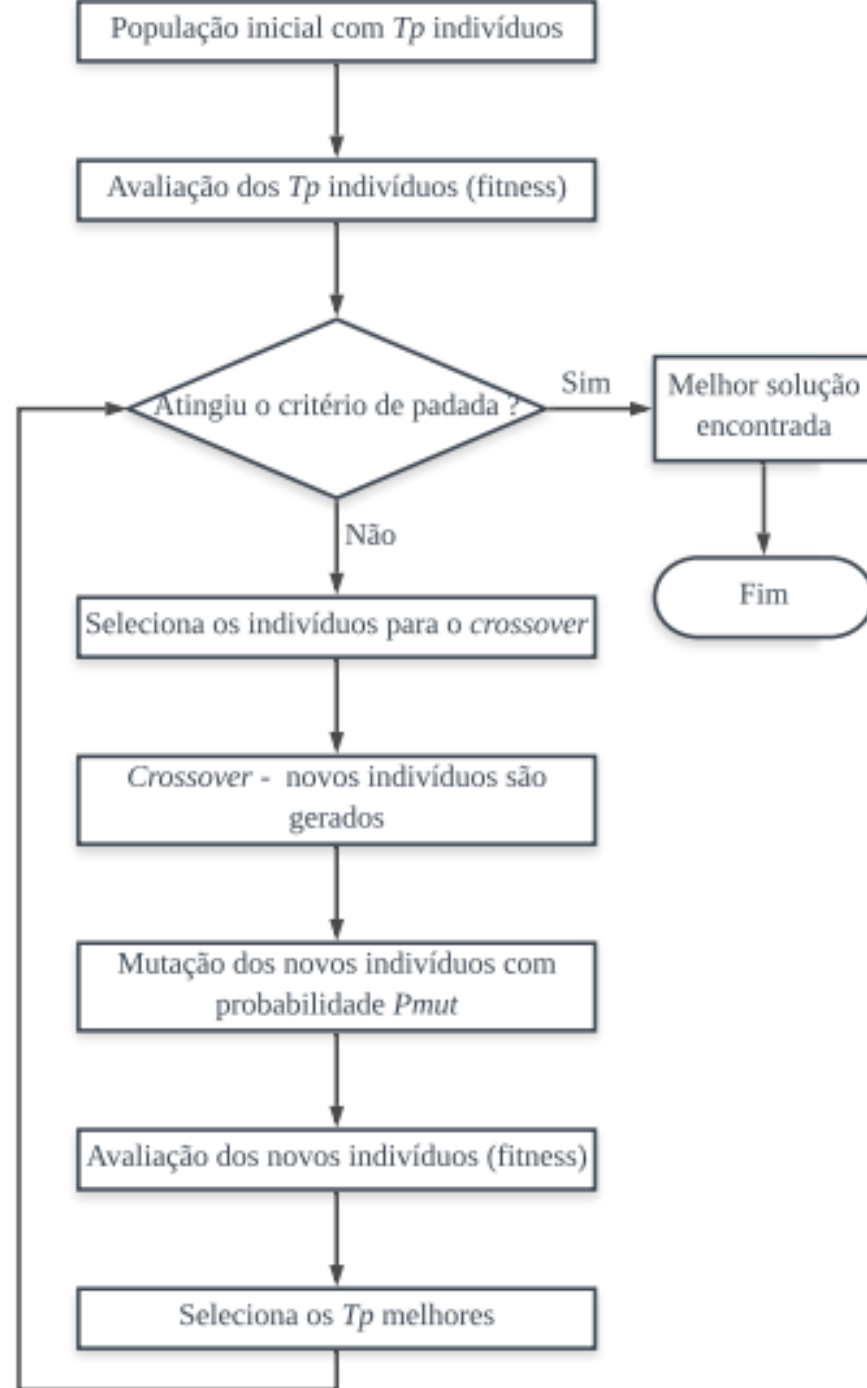
- Preferencialmente utilizados em problemas onde o **espaço de busca é muito grande**
- Não necessariamente (quase nunca) queremos chegar na solução **ótima**, mas sim **naquela mais próxima da ótima (melhor possível)**

Algoritmos Genéticos

- Os elementos principais são:
 - População de indivíduos (codificados para representarem as possíveis soluções);
 - Avaliação dos indivíduos (Aptidão ou Fitness);
 - Seleção (Quem irá propagar os genes?);
 - Recombinação (crossover);
 - Mutação

Algoritmos Genéticos

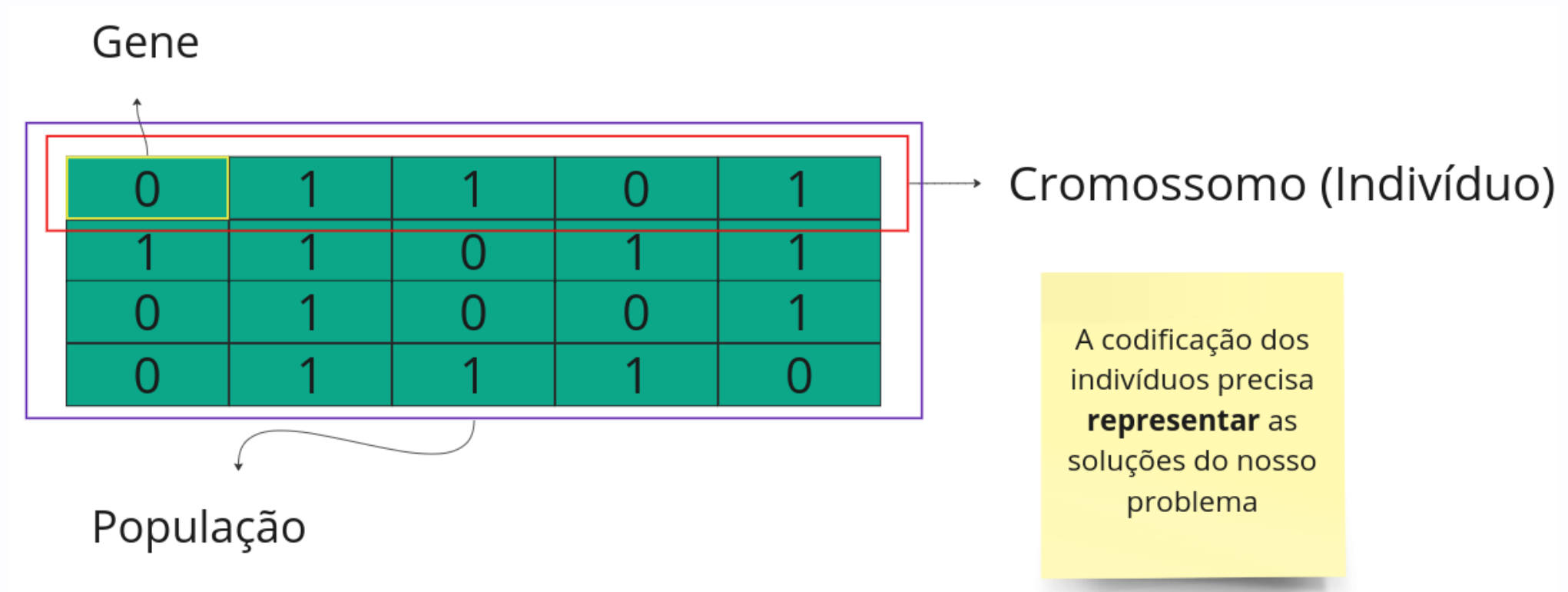
- A aplicação desses **operadores genéticos** promove uma competição entre os indivíduos
- Sobrevivência dos mais aptos



Cromossomo - AGs

- Assim como na biologia, em AGs **codificamos informação**
- Em AGs, os termos **cromossomos e indivíduos são intercambiáveis**
- Representam **possíveis soluções**
- Forma mais básica é o **string de bits**

Estrutura de um Cromossomo - AGs



Problema da mochila - Cromossomo

- Clássico da otimização combinatória
- Enunciado simples mas solução complexa
- **Encher uma mochila de peso X com n itens, maximizando o valor total desses itens**
- **Restrição:** Peso da mochila não deve ser excedido

Problema da mochila - Cromossomo

- Número de combinações pode crescer exponencialmente conforme o número de itens
- Para 5 itens temos $2^5 = 32$ combinações possíveis
- Para 12 itens $2^{12} = 4.096$ combinações possíveis
- Quanto **tempo** levaria para encontrar a melhor solução?
- Computacionalmente pode ser custoso comparar tantas combinações

**Como representar soluções desse
problema em AGs?**

Como codificar nossos indivíduos?

0 = Levar

1 = Não levar



Aptidão: Valor total dos itens

Genes = Item 1 Item 2 Item 3 Item 4 Item 5

```
{  
  "nome": exemplo,  
  "peso": 5.0,  
  "valor": 100  
}
```

Problema da mochila - Cromossomo

- Usamos string de bits
- Cada **gene (posição na string)** representa um item
- Esses itens possuem suas **propriedades**
- Para a **aptidão (fitness)** podemos usar o valor **total dos itens** que vamos levar (maximizar valor)

**Em AGs precisamos definir um critério
de parada**

**No exemplo da mochila, qual poderia
ser nosso critério de parada?**

Problema da mochila - Cromossomo

- Para o problema da mochila, onde temos um número de combinações muito grande, **o valor máximo pode ser desconhecido**
- Portanto, não conseguimos usar esse valor como critério de parada (Não sabemos a resposta)
- Por esse motivo, podemos usar o **número de gerações**
- **Obs: Passamos a ter um parâmetro extra**

Parâmetros dos AGs

- Alguns parâmetros que precisamos testar com os AGs:
 - TP = Tamanho da População
 - P_{mut} = Probabilidade de mutação
 - P_{cross} = Probabilidade de crossover
 - Número de gerações (iterações)

Operador de Seleção

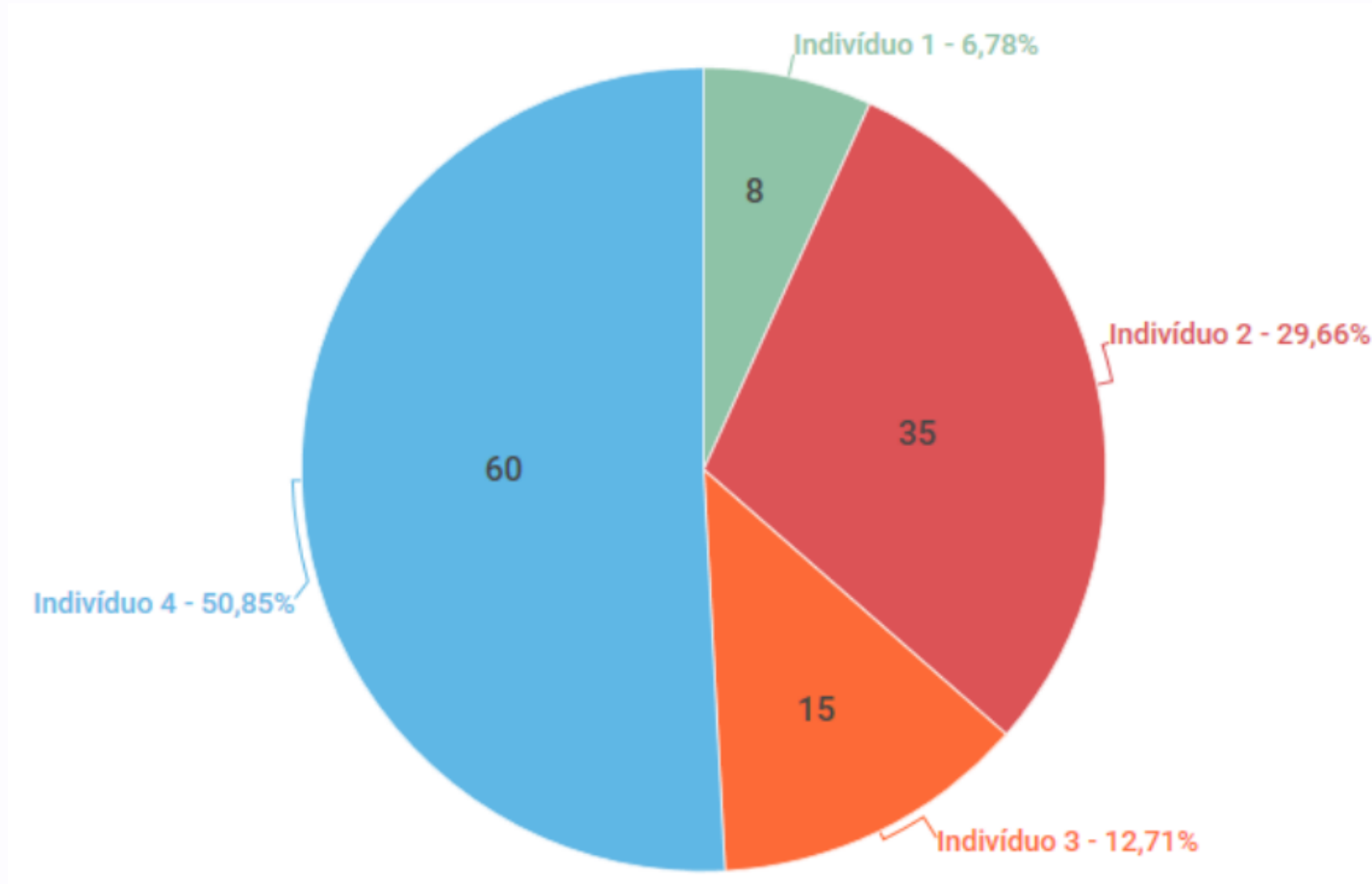
- Simula a sobrevivência do mais apto
- Baseado no fitness, seleciona quem vai reproduzir e perpetuar seus genes
- De maneira geral seleciona os mais aptos com mais frequência
- Mas **não descarte os menos aptos**

Operador de Seleção - Roleta

- Simula uma roleta
- Cada indivíduo possui uma "porção" da roleta
- Tamanho da porção depende do fitness
- Menos aptos também pode ser selecionados

$$p_i = \frac{f_i}{\sum_{i=1}^N f_i}$$

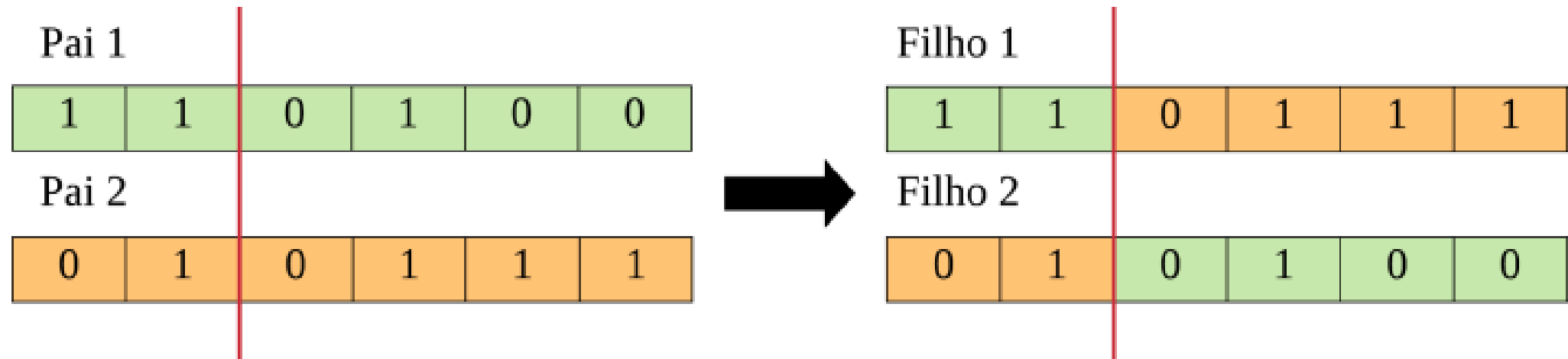
Operador de Seleção - Roleta



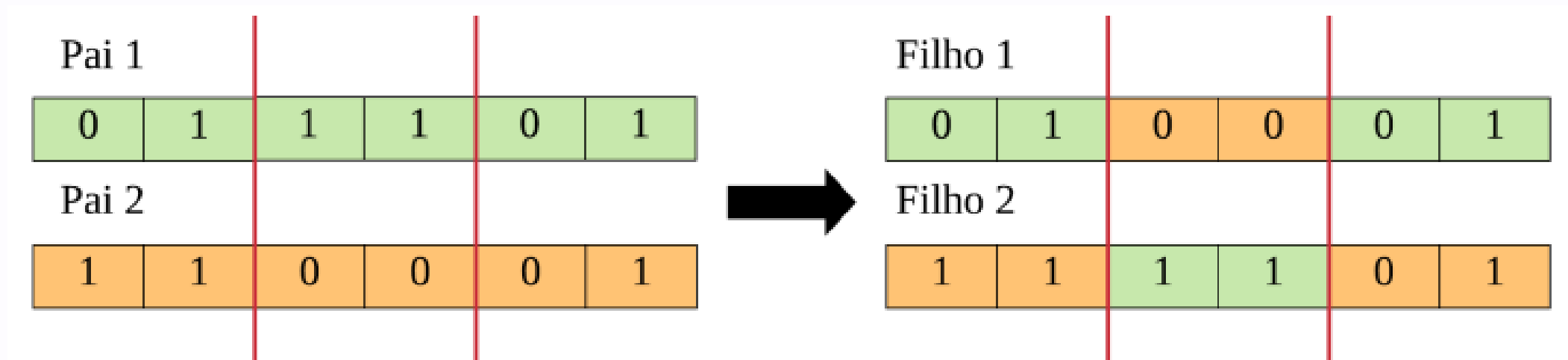
Operador de Crossover

- Indivíduos escolhidos pelo operador de seleção são conhecidos como **pais**
- São neles que o Crossover será aplicado e novos **filhos será gerados para a próxima geração**
- Simula a reprodução sexuada para a **troca de informação genética**

Operador de Crossover - 1 ponto



Operador de Crossover - 2 pontos



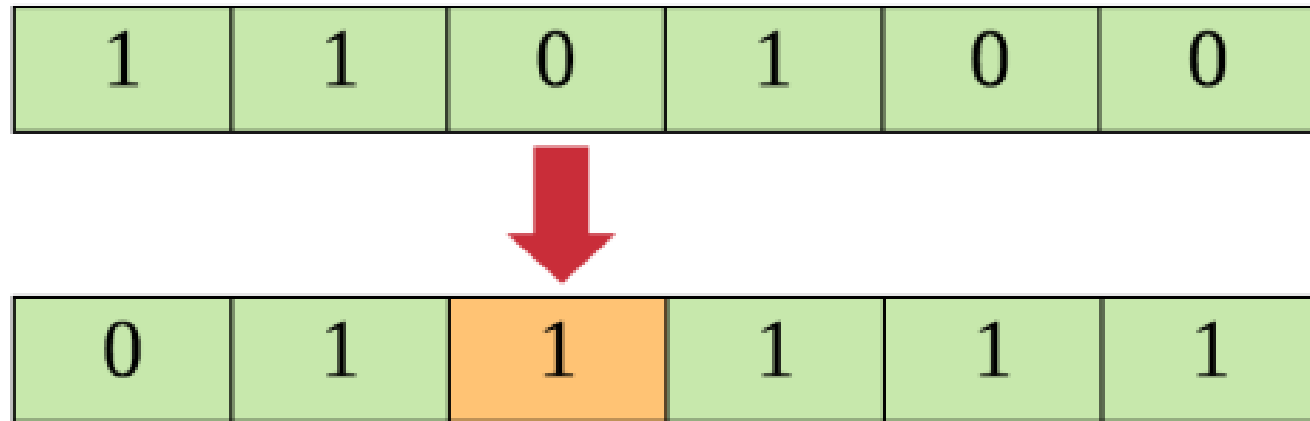
Operador de Mutação

- O **elitismo** pode prejudicar a capacidade de busca dos AGs
- Ao longo das gerações podemos ter indivíduos muito parecidos
- Fenômeno dos **mínimos locais/ótimos locais**
- Precisamos introduzir variabilidade genética
- Isso é feito pela seleção, mas principalmente por **mutação**

Operador de Mutação

- Selecionamos randomicamente os valores de um ou mais genes de um indivíduo para modificar
- A seleção é feita baseada no parâmetro P_{mut} (probabilidade de mutação)
- Para codificação binária de string de bits, basta utilizar um **bit swap**

Operador de Mutação



Reinserção

- Como será selecionados os indivíduos para a próxima geração?
- É a última etapa
- Reinserção pura: Apenas os filhos passam para a próxima geração
- Melhores pais e filhos: Os T_p da população corrente passam para a próxima geração
- Elitismo: Os n melhores da geração anterior passam para a próximo. **O melhor da próxima geração é pelo menos igual ao da anterior**