

Inteligência Artificial

II

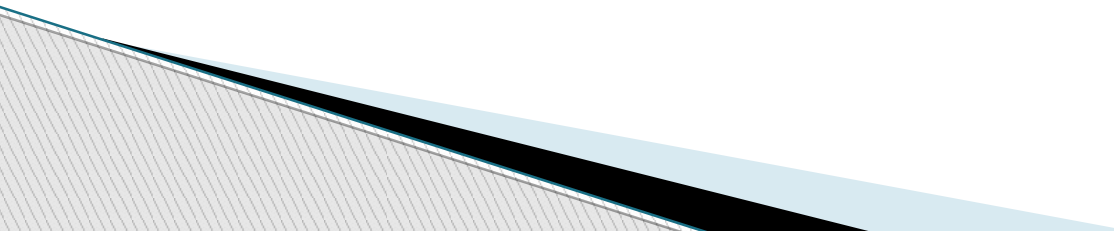
Algoritmos Genéticos

Prof. Tales Bitelo Viegas

<http://facebook.com/ProfessorTalesViegas>

The bottom of the slide features a decorative graphic consisting of several overlapping, wavy lines in shades of gray and light blue, creating a modern, abstract background.

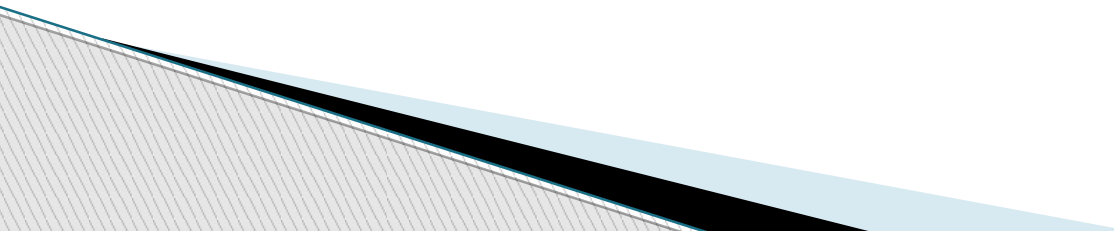
Algoritmos Genéticos

- ▶ Pode ser usado quando não é necessário gerar os estados intermediários (caminho) do estado inicial ao final (objetivo)
 - ▶ São métodos de otimização e busca inspirados nos mecanismos de evolução de populações dos seres vivos
 - ▶ Foram introduzidos por John Holland (1975) e popularizados por um dos seus alunos, David Goldberg (1989)
- 

Algoritmos Genéticos

- ▶ Seguem o princípio da seleção natural e sobrevivência (Charles Darwin – livro “A Origem das Espécies”, 1859)
 - “Quanto melhor um indivíduo se adaptar ao seu meio ambiente, maior será a sua chance de sobreviver e gerar descendentes”.

Algoritmos Genéticos

- ▶ O algoritmo começa com um **conjunto de soluções** (representados por **cromossomas**) chamado **população**.
 - ▶ Soluções de uma população são utilizadas para formar uma nova população (**reprodução**). Isto é motivado pela esperança que a nova população será melhor que a primeira.
- 

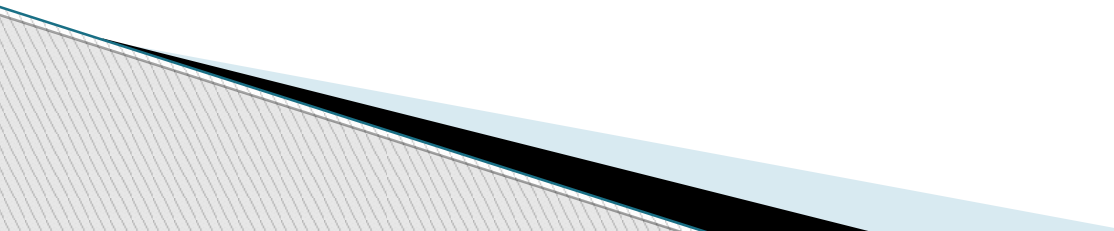
Algoritmos Genéticos



Algoritmos Genéticos



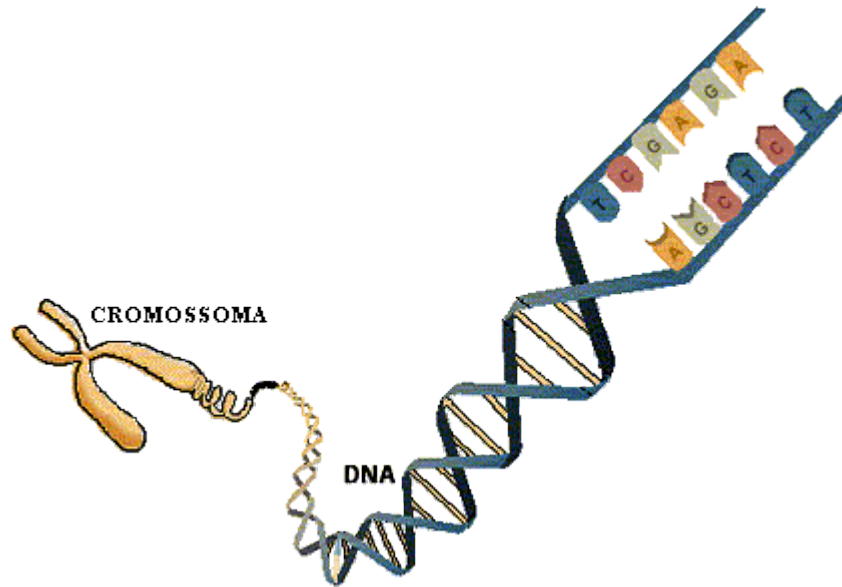
Algoritmos Genéticos

- ▶ Soluções são selecionadas para formar novas gerações de soluções de acordo com a sua adequação – quanto melhores, mais chances de reprodução terão.
 - ▶ Este processo é repetido até que alguma condição é satisfeita (por exemplo, o número de populações ou o aperfeiçoamento da melhor solução)
- 

Algoritmos Genéticos

► Cromossomo

- Consistem de genes – sequências de DNA – que servem para determinar as características de um indivíduo.



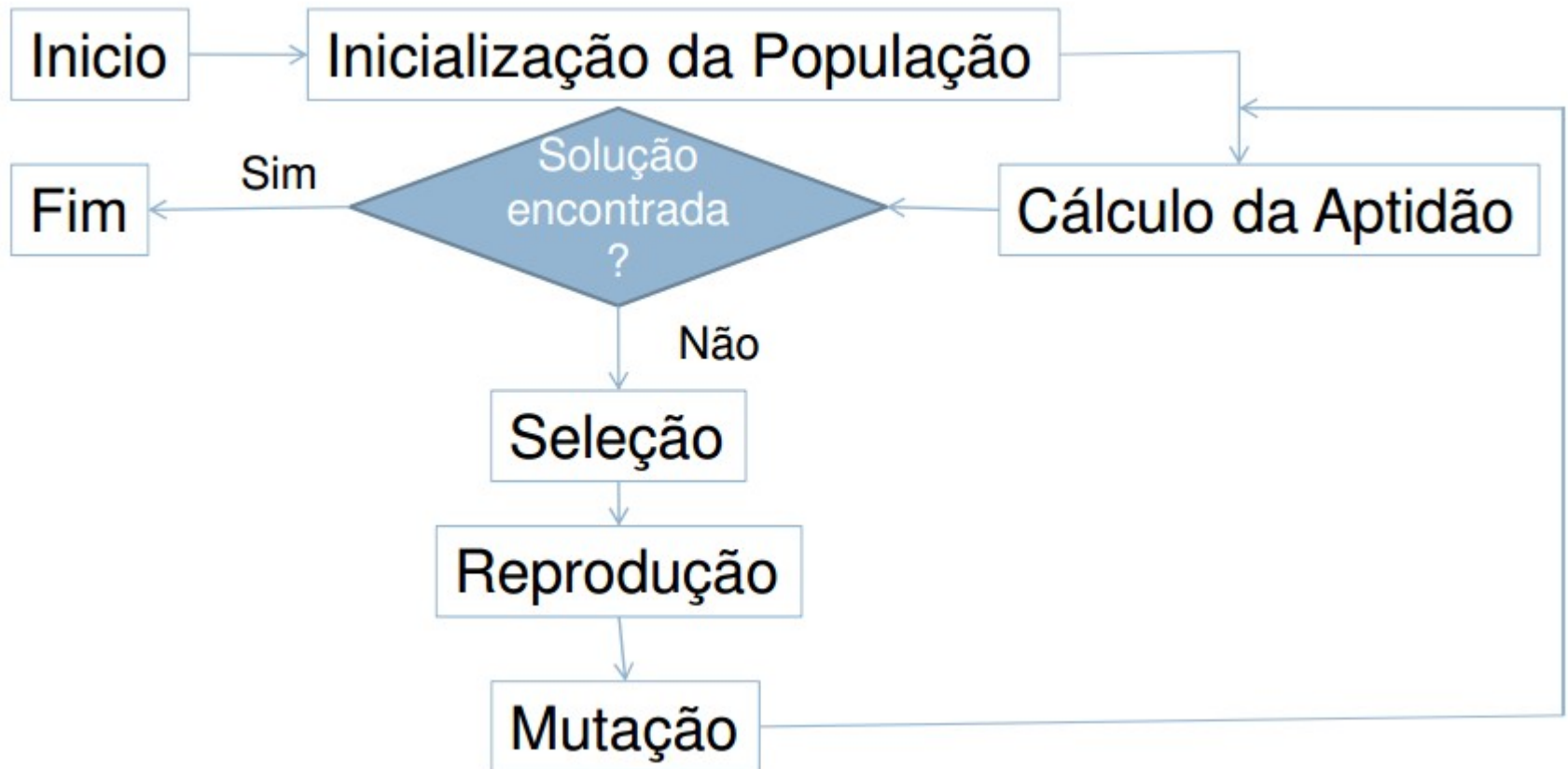
Algoritmos Genéticas

▶ Reprodução

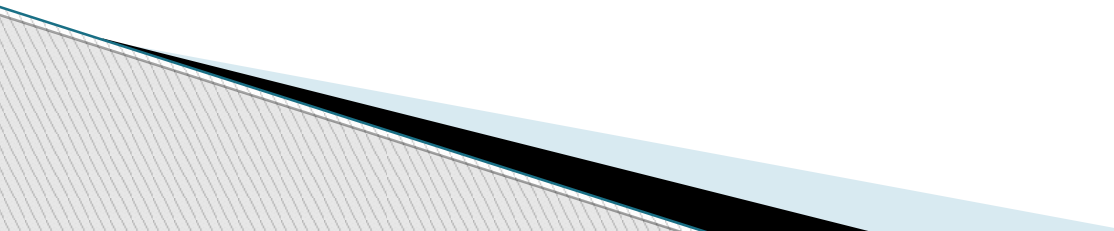
- Durante o processo de reprodução ocorre a recombinação (ou **crossover** – cruzamento). Genes dos pais se combinam para formar novos cromossomos.
- Os descendentes criados podem sofrer **mutações**, ou seja, os elementos do DNA podem ser trocados
- A **adaptação** de um organismo pode ser medida pelo sucesso do mesmo em sua vida

Algoritmos Genéticos

► Estrutura Básica



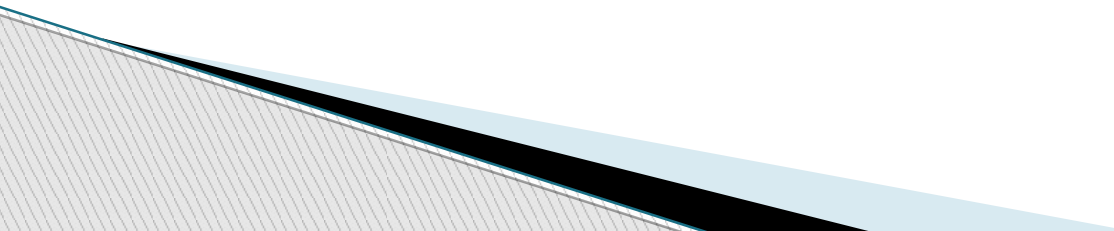
Algoritmos Genéticos

- ▶ Cada iteração do algoritmo genético corresponde à aplicação de um conjunto de quatro operações básicas:
 - Cálculo de aptidão
 - Seleção
 - Cruzamento
 - Mutação
- 

Algoritmos Genéticos

- ▶ Ao fim destas operações cria-se uma nova população chamada de **geração** que, espera-se, represente uma melhor aproximação da solução do problema de otimização que a população anterior

Algoritmos Genéticos

- ▶ A população inicial é gerada atribuindo-se aleatoriamente valores aos genes de cada cromossomo
 - ▶ A aptidão de um indivíduo da população é medida por uma função chamada de função objetivo do problema de otimização
 - ▶ Como critérios de parada do algoritmo, em geral, são usados a aptidão do melhor indivíduo do grupo em conjunto com a limitação do número de gerações
- 

Algoritmos Genéticos

- ▶ Para usar algoritmos genéticos é necessário
 - Codificar o problema
 - Definir a função de aptidão
 - Determinar o tamanho da população
 - Escolher a estratégia da solução
 - Implementar o algoritmo, usando os operadores genéticos adequados à forma de codificação escolhida.

Algoritmos Genéticos

- ▶ A codificação pode ser:
 - Binária
 - ▢ É a mais comum devido a sua simplicidade
 - ▢ Cada cromossomo é uma String de bits – 0 ou 1
 - ▢ Crom A = 1 0 1 1 0 0 1 0 1 1
 - ▢ Crom B = 1 1 1 1 1 1 0 0 0 0
 - ▢ Exemplo de uso: problema da mochila
 - ▢ Codificação: cada bit diz se um elemento está ou não na mochila

Algoritmos Genéticos

- ▶ A codificação pode ser por valor:
 - Usado em problemas onde valores mais complicados são necessários
 - Cada cromossomo é uma sequência de valores
 - ▢ Crom A: 1.2324 5.3242 0.4556 2.3293 2.4545
 - ▢ Crom B: ABDJEIFJDHDIERJFDLDFLFEGT
 - ▢ Crom C: 4, 8, 15, 16, 23, 42
 - ▢ Crom D: (up), (up), (back), (back), (left), (right), (left), (right)

Algoritmos Genéticos

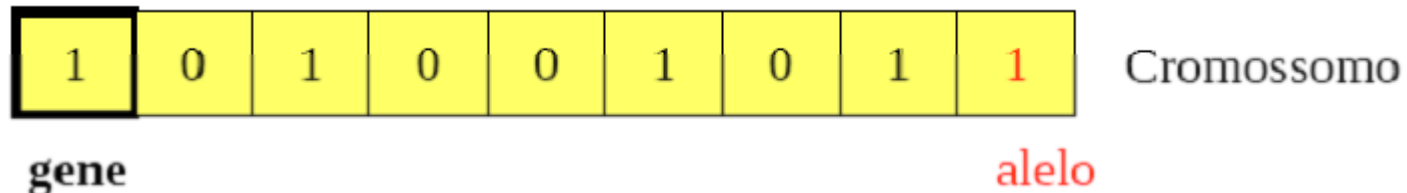
- ▶ A codificação pode ser por valor:
 - Exemplo de uso: dada uma estrutura, encontrar pesos para uma rede neural
 - Codificação: valores reais num cromossomo representam pesos em uma rede neural

Codificação - Exemplo

- ▶ **Problema:**
- ▶ Separe o conjunto C de valores inteiros a seguir em dois subconjuntos C1 e C2, de forma que a soma dos elementos de C1 seja próxima ou igual à soma dos elementos de C2
 - $C = \{ 10, 20, 1, 2, 10, 5, 5, 3, 10 \}$

Codificação - Exemplo

- ▶ Exemplo de codificação em binário
 - $C = \{ 10, 20, 1, 2, 10, 5, 5, 3, 10 \}$



- Cada cromossomo representa uma solução possível para o problema
- De acordo com a codificação:
 - ▢ $C1 = \{ 10, 1, 5, 3, 10 \}$
 - ▢ $C2 = \{ 20, 2, 10, 5 \}$

Codificação - Exemplo

- ▶ Função de aptidão
 - **$\text{abs}(\text{soma}(\mathbf{C1}) - \text{soma}(\mathbf{C2}))$** , onde soma é uma função que soma os elementos do conjunto passado como parâmetro. Quanto mais próximo de zero, melhor será a solução
- ▶ Exemplo:
 - $\mathbf{C1} = \{ 10, 1, 5, 3, 10 \}$
 - $\mathbf{C2} = \{ 20, 2, 10, 5 \}$
 - $F = \text{abs}(29 - 37) = 8$

Tamanho da População

- ▶ Número de soluções candidatas (cromossomos) que serão consideradas a cada iteração do algoritmo
- ▶ Exemplo: 4 cromossomos

0	1	0	1	0	0	1	0	1	1
1	0	0	1	1	0	1	0	0	1
2	0	0	0	1	0	0	1	1	0
3	1	1	0	0	0	0	1	0	1

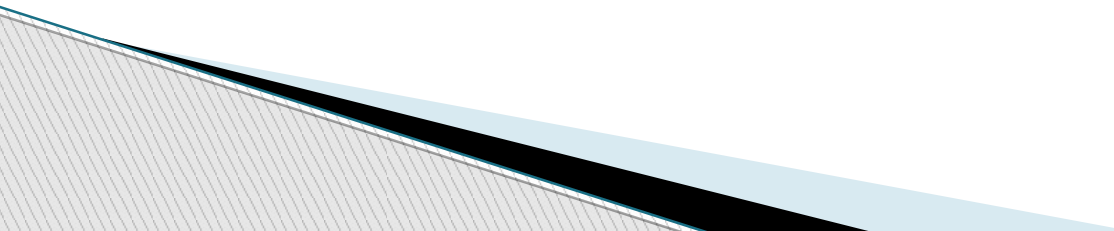
Estratégia de Seleção

- ▶ **Elitismo:** passa o melhor cromossomo diretamente para a geração seguinte
- ▶ Outras estratégias:
 - **Torneio:** escolhe randomicamente 2 candidatos e pelo valor da função de aptidão escolhe o melhor
 - $C = \{ 10, 20, 1, 2, 10, 5, 5, 3, 10 \}$

0	1	0	1	0	0	1	0	1	1
3	1	1	0	0	0	0	1	0	1

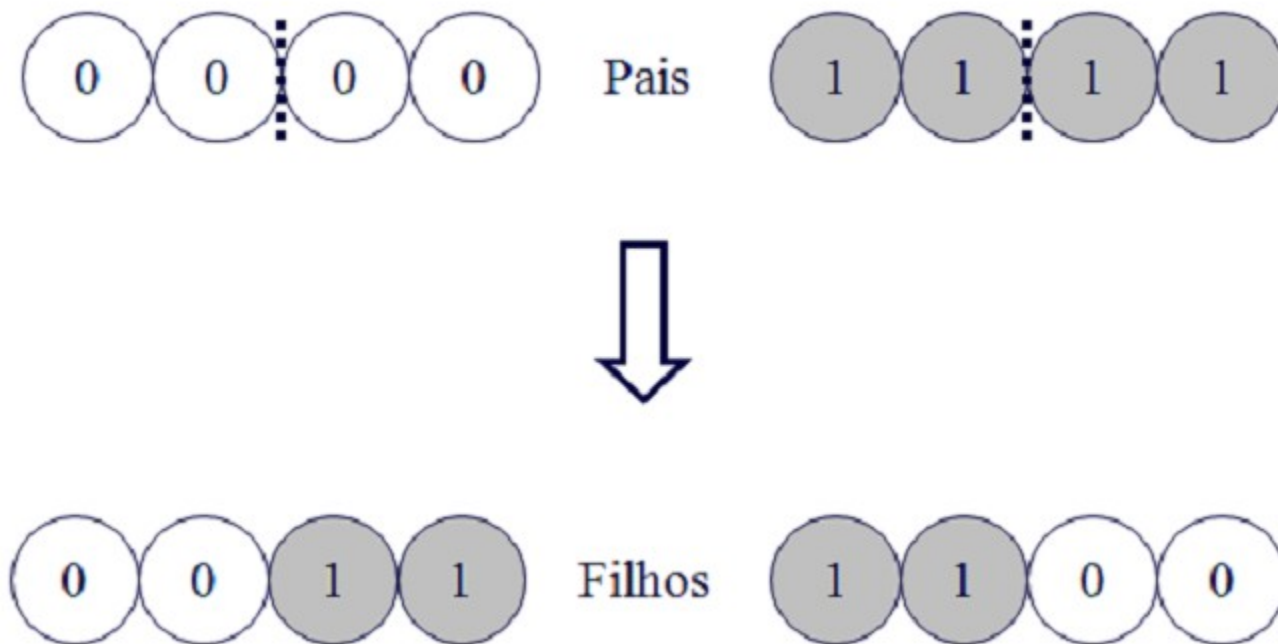
- Qual das 2 soluções será considerada a melhor?

Crossover

- ▶ Combina as informações genéticas de dois indivíduos (pais) para gerar novos indivíduos (filhos)
 - ▶ Versões mais comuns criam sempre dois filhos para cada operação
- 

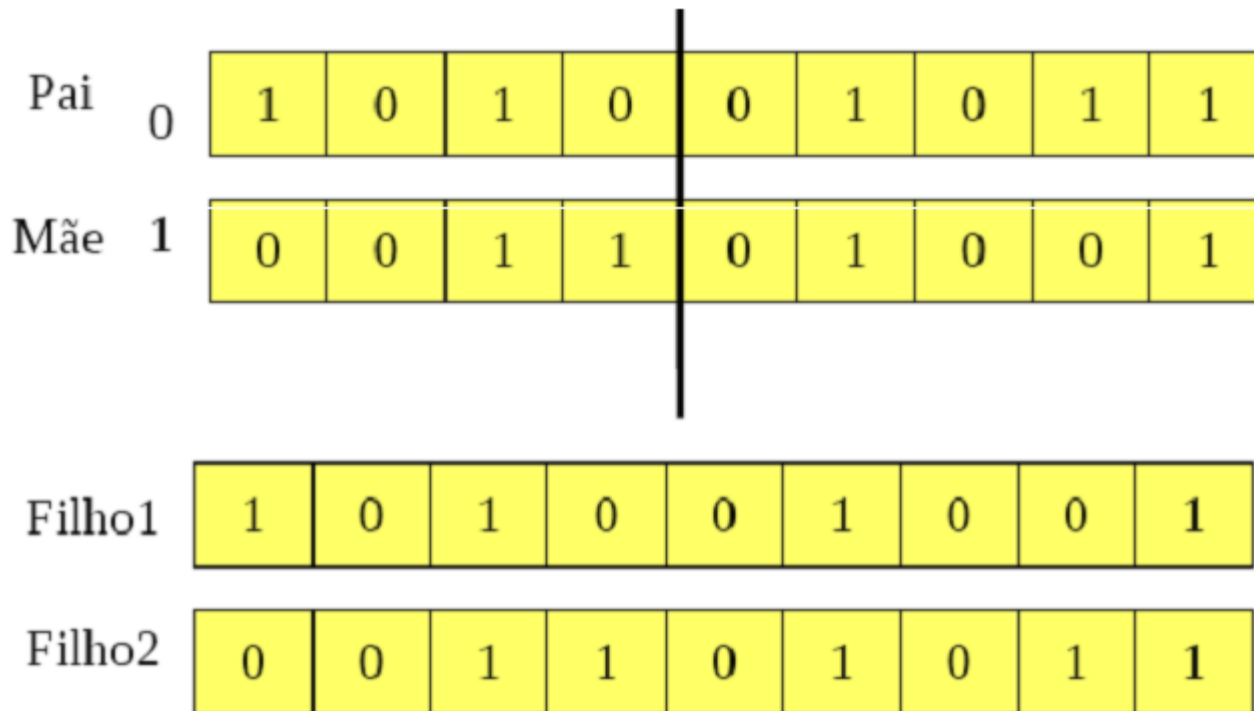
Crossover

- ▶ Crossover (cruzamento) – Um ponto



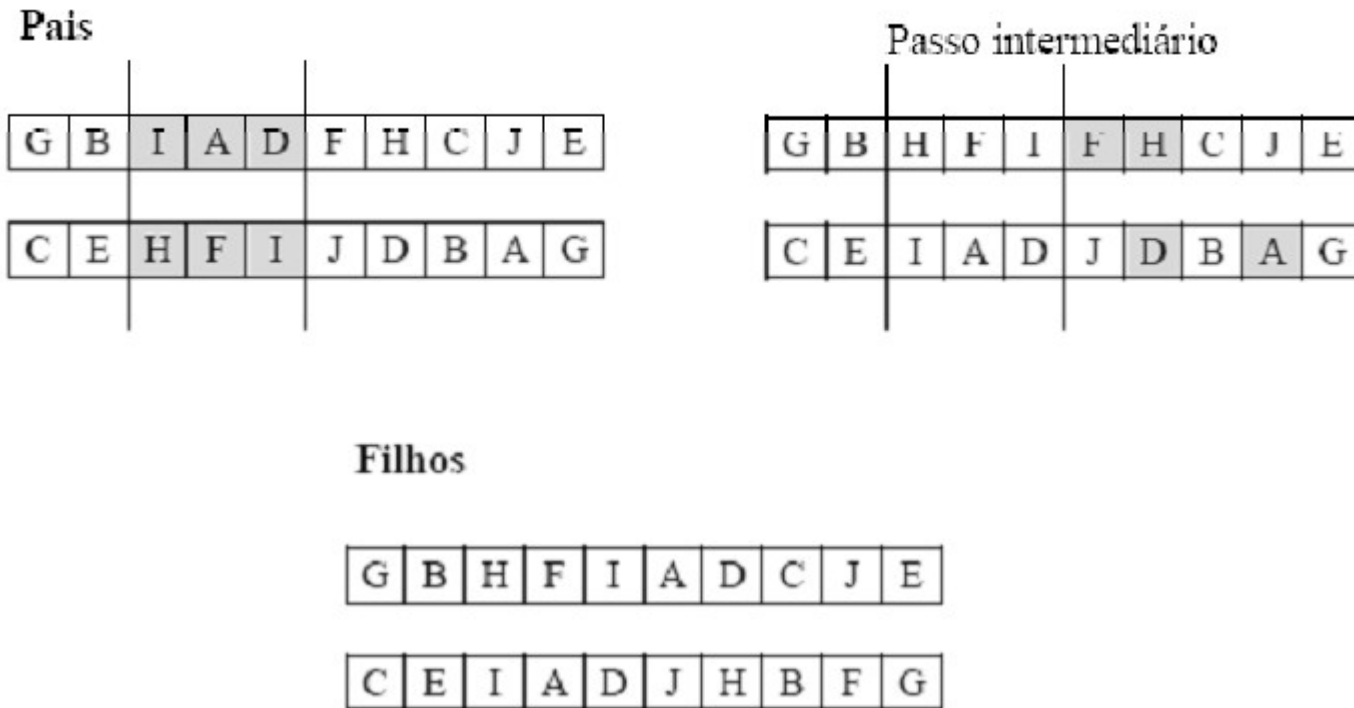
Crossover

- Crossover (cruzamento) – Um ponto



Crossover

- Crossover (cruzamento) – Dois pontos



Crossover

► Crossover uniforme

- Os bits são copiados OU do primeiro cromossomo OU do segundo, de acordo com uma **máscara de cruzamento**
- Quando o valor da máscara for 1, copia-se o valor do **primeiro** cromossomo
- Quando o valor da máscara for 0, copia-se o valor do **segundo** cromossomo
- Para o segundo filho, inverte-se a máscara

Crossover

- ▶ Pai1: 10001000
- ▶ Pai2: 10110010
- ▶ Máscara: 10010011
- ▶ Filho1: 10100000
- ▶ Filho2: 10011010

Mutação

- ▶ Depois que um cruzamento é realizado, acontece a mutação
- ▶ A mutação tem a intenção de prevenir que todas as soluções do problema dessa população caiam em um ponto ótimo local
- ▶ A operação de mutação muda aleatoriamente a descendência criada pelo cruzamento
- ▶ No caso de uma codificação binária, podemos mudar aleatoriamente alguns bits escolhidos de 1 para 0, ou de 0 para 1

Mutação

► Exemplo:

- Filho1 antes: 1101111000011110
- Filho2 antes: 1101100100110110
- Filho1 depois: 1100111000011110
- Filho2 depois: 1101101100110100

Resumindo o Algoritmo

- ▶ **[Início]** Gere uma população aleatória de n cromossomas (soluções adequadas para o problema)
- ▶ **[Adequação]** Avalie a adequação $f(x)$ de cada cromossoma x da população
- ▶ **[Nova População]** Cria uma nova população repetindo os passos seguintes até que a nova população esteja completa:
 - **[Seleção]** selecione de acordo com a sua adequação (melhor adequação, mais chances de ser selecionado) dois cromossomas para serem os pais
 - **[Cruzamento]** com a probabilidade de cruzamento, cruze os pais para formar a nova geração. Se não realizar cruzamento, a nova geração será uma cópia exata dos pais
 - **[Mutação]** com a probabilidade de mutação, altere os cromossomas da nova geração nos locus (posição dos cromossomas)
 - **[Aceitação]** Coloque a nova descendência na nova população
- ▶ **[Substitua]** Utilize a nova população gerada para a próxima rodada do algoritmo
- ▶ **[Teste]** Se a condição final for atingida, **pare** e retorne a melhor solução da população atual
- ▶ **[Repita]** Vá para a Adequação!

Aplicações

- ▶ Otimizações de Funções Matemáticas
 - ▶ Otimização de Planejamento
 - ▶ Problema de otimização da Rota de Veículos
 - ▶ Otimização de Layout de Círculos
 - ▶ Otimização de Distribuição
- 