

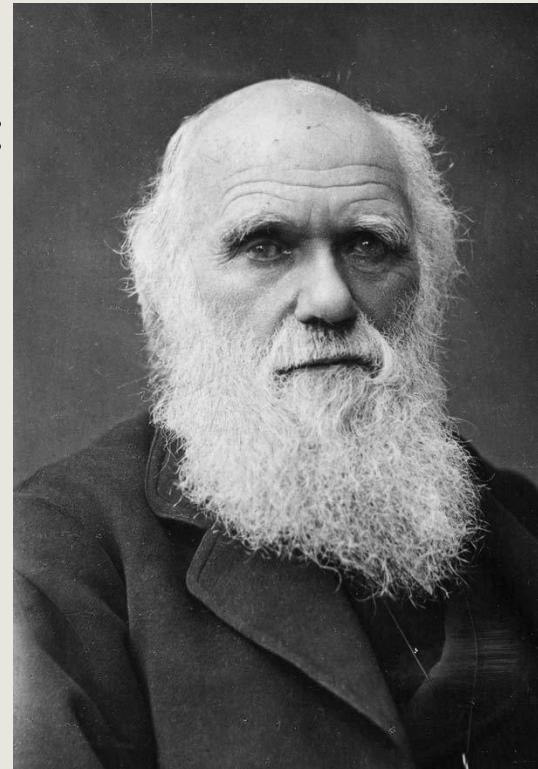
# ALGORITMOS GENÉTICOS E REDES NEURAIS

## ALGORITMOS GENÉTICOS

Prof. Flávio Belizário da Silva Mota  
Universidade do Vale do Sapucaí – UNIVAS  
Sistemas de Informação

# COMPUTAÇÃO EVOLUCIONÁRIA

- A **computação evolucionária** é uma sub-área da Inteligência Artificial, na qual diversos **algoritmos evolucionários** estão presentes. Esses algoritmos se baseiam na **teoria da evolução das espécies** descrita por Charles Darwin em 1859.
- Os algoritmos evolucionários apresentam **quatro elementos básicos**:
  - Uma **população de indivíduos**, onde cada indivíduo corresponde a uma **solução candidata** do sistema
  - Uma maneira de **criar novos indivíduos**, a partir de indivíduos já **existentes**
  - Uma forma de **medir a qualidade da solução** que cada indivíduo corresponde
  - Um método de **selecionar os melhores indivíduos**, aplicando assim o princípio da seleção natural



# ALGORITMOS GENÉTICOS



- Algoritmos Genéticos (AGs) são algoritmos computacionais que **buscam uma melhor solução** para um determinado problema, através da **evolução de populações de soluções codificadas através de cromossomos artificiais**.
- Muitos problemas reais são **difíceis para métodos determinísticos**.
- Espaços de busca **grandes, não lineares ou com múltiplos ótimos locais**.
- AGs realizam uma busca **paralela e adaptativa**.
- Muito usados em engenharia, logística, otimização, finanças e pesquisa operacional.

# ALGORITMOS GENÉTICOS

Um AG trabalha com uma população de soluções candidatas, ou seja, um conjunto de possíveis soluções para um problema. Em cada geração:

1. Avalia os indivíduos por meio de uma função de aptidão.
2. Seleciona os mais aptos.
3. Recombina (crossover) partes de seus cromossomos.
4. Aplica mutações para preservar diversidade.
5. Forma uma nova geração mais adaptada.

Com o tempo (as gerações), a população tende a convergir para boas soluções.

# ALGORITMOS GENÉTICOS

Para entender melhor os AGs, podemos separá-los nos seguintes componentes:

1. Problema a ser otimizado
2. Codificação do cromossomo
3. Avaliação
4. Seleção
5. Operadores Genéticos

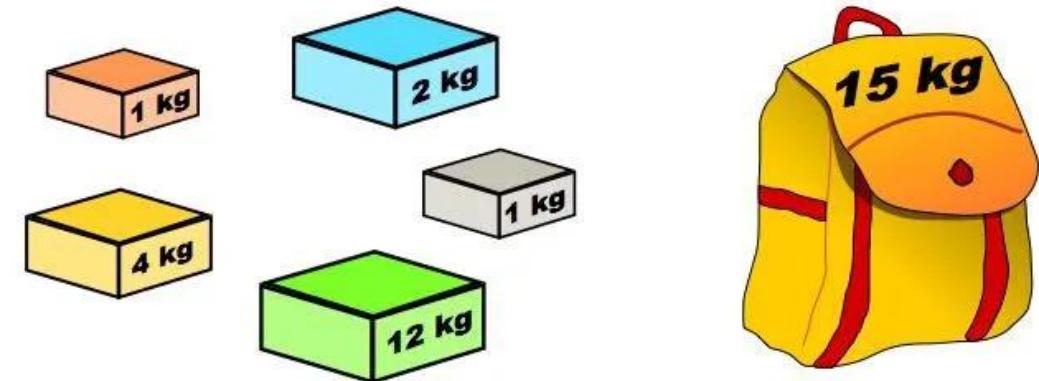
# PROBLEMAS A SEREM OTIMIZADOS

Geralmente, AGs são aplicados em **problemas de otimização**:

- Problemas com diversos parâmetros ou características que precisam ser **combinadas em busca da melhor solução**.
- Problemas com **muitas restrições ou condições** que não podem ser representadas matematicamente.
- Problemas com **grandes espaços de busca**.

Um exemplo é o problema da mochila (Knapsack problem). Podemos entendê-lo como **o desafio de encher uma mochila sem ultrapassar um determinado limite de peso, otimizando o valor do produto carregado**.

A **solução exata** desse problema requer a avaliação de todas as combinações possíveis de itens para encontrar a mais vantajosa. Para  $n$  itens, existem  $2^n$  combinações possíveis, o que leva a uma complexidade de tempo de  $O(2^n)$ .



# CODIFICAÇÃO DO CROMOSSOMO

- No contexto dos AGs, os **indivíduos (soluções)** são representados por **cromossomos** e cada posição do cromossomo representa uma **característica do problema**, e é chamado de **gene**. Os genes por sua vez armazenam os **valores** da caracterísca nos **alelos**.
- O processo de **codificação de indivíduos** é onde ocorre a representação de cada possível solução de um problema como uma sequênciā de símbolos.
- A representação do cromossoma **depende do tipo de problema** e do que, essencialmente, se deseja manipular geneticamente.
- Os principais tipos de representação são a **codificação binária** e a **codificação real**.

# CODIFICAÇÃO BINÁRIA

- Em grande parte dos problemas práticos tratados com AGs é utilizada a codificação binária, pois geralmente as variáveis do problema podem ser representadas por números binários.
- Um indivíduo é representado usando um **vetor binário que tem valores 0 e 1**. O número de bits necessários para representar o indivíduo varia de acordo com o intervalo de valores possíveis no espaço de busca.
- Considere o problema de encontrar o **valor máximo de uma função  $f(x) = x^2$** , com  $x$  sendo um valor inteiro entre 0 e 63. A representação binária de soluções precisaria usar um cromossoma de 6 bits, pois:

C1 -> 0 0 0 0 0 0 -> representa  $x=0$

C2 -> 1 1 1 1 1 1 -> representa  $x=63$

# CODIFICAÇÃO BINÁRIA

- Já para representar valores reais, o tamanho do cromossomo dependerá de quantas casas decimais de precisão a solução exige. Considerando  $t$  o tamanho do intervalo de valores e  $p$  a precisão, temos a seguinte equação que define o tamanho do cromossomo:

$$n = \lfloor \log_2(t \times 10^p) \rfloor + 1$$

- Por exemplo, se tivermos um intervalo entre 1 e 3,  $t = 2$  ( $t=3-1=2$ ). Se quisermos uma precisão de 5 casas decimais ( $p=5$ ), então seria necessário um cromossomo de 18 bits ( $n = \lfloor \log_2(2 \times 10^5) \rfloor + 1 = 18$ ).
- Para mapear o valor da variável no intervalo binário, aplicamos a função abaixo e depois convertemos o valor para a base binária, sendo que  $x$  é o valor real da variável e  $\hat{x}$  o valor mapeado.

$$\hat{x} = \frac{(x - x_{\min}) \times (2^n - 1)}{t}$$

# CODIFICAÇÃO BINÁRIA

- Para decodificar um cromossomo, é necessário convertê-lo da base 2 para a base 10, produzindo um valor discreto.

$$s = [b_n b_{n-1} \dots b_2 b_1 b_0] = \sum_{i=0}^n b_i \times 2^i = \hat{x}$$

- Posteriormente, o valor convertido é mapeado de volta ao espaço de busca, sendo  $x_{\min}$  e  $x_{\max}$  os limites do intervalo desse espaço.

$$x = x_{\min} + \left( \frac{x_{\max} - x_{\min}}{2^n - 1} \right) \times \hat{x}$$

# CODIFICAÇÃO REAL

- A codificação binária não é tão adequada quando o problema tem **alta dimensionalidade ou necessita de alta precisão numérica**.
- Na codificação real, o cromossomo é representado por um vetor de números reais, onde cada gene contém o próprio valor da variável do problema.

1.029	0.215	0.984	0.128	1.854	1.024
-------	-------	-------	-------	-------	-------

- Os cromossomos gerados são menores em relação a codificação binária e são mais fáceis de compreender por um ser humano. Entretanto, computacionalmente, gastam mais espaço em memória.

# AVALIAÇÃO – FUNÇÃO DE APTIDÃO

- A avaliação é o que conecta o AG ao mundo externo. Ela tem como objetivo **fornecer uma medida de aptidão (fitness) de cada indivíduo da população**, o que orienta o processo de busca.
- A avaliação é feita através de uma função que melhor representa o problema, sendo específica de cada problema.
- Por exemplo, considere o problema de maximizar  $f(x) = x^2$ , ou seja, obter **o maior valor possível para a função**. Se considerarmos os indivíduos C1 e C2 abaixo, C1 é **mais apto** que C2, pois apresenta um valor resultante de  $f(x)$  maior (**maximiza a função**).

Indivíduo	Cromossomo	x	f(x)
C1	0 0 1 0 0 1	9	81
C2	0 0 0 1 0 0	4	16

# SELEÇÃO

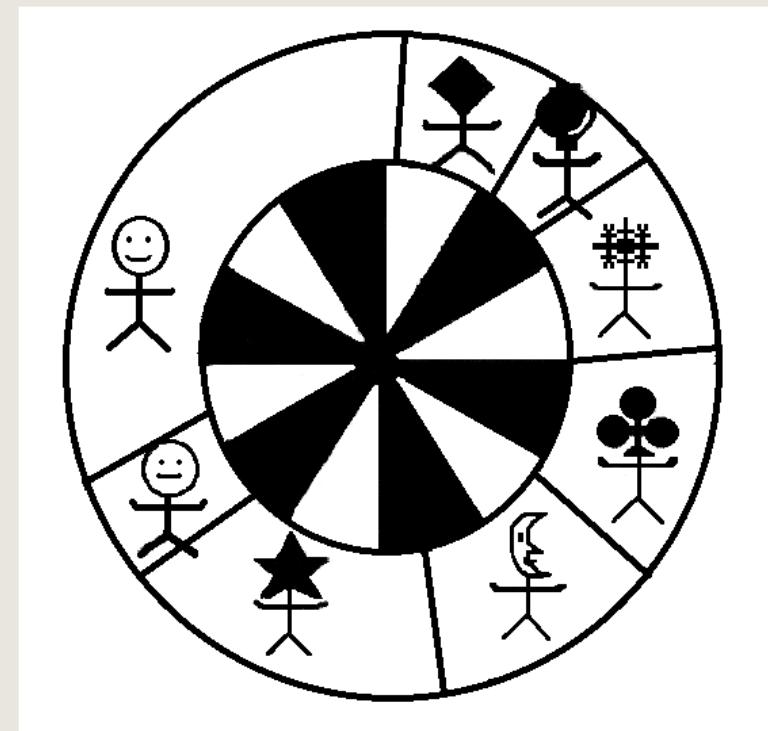
- A seleção tem como objetivo **escolher os melhores indivíduos da população**, emulando o processo de seleção natural.
- A seleção é baseada na **aptidão (fitness)** dos indivíduos: os mais aptos têm maior probabilidade de serem escolhidos para se reproduzirem (crossover).
- Existem diversos métodos de seleção, mas a grande maioria deles utiliza a **proporção do valor de aptidão do indivíduo em relação a toda população**, dado por:

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

Onde  $f_i$  é a aptidão do indivíduo e  $N$  é o tamanho da população.

# SELEÇÃO – MÉTODO DA ROLETA

- O método mais utilizado é o método da roleta. A ideia é que uma roleta fictícia armazena os indivíduos em cada casa da roleta, sendo que o tamanho da casa é proporcional ao valor da aptidão (da probabilidade) do indivíduo.
- Assim, indivíduos com maior aptidão têm maior chance de serem escolhidos para gerar descendentes.
- Esse método pode ser considerado muito “agressivo”, pois discrimina indivíduos com menor valor de aptidão, o que em alguns casos pode levar a solução a ficar presa em um **máximo local**.



# RANK, TORNEIO E ELITISMO

- O método de seleção por rank **ordena os indivíduos pelo valor da aptidão e depois são alocadas probabilidades de seleção de acordo com o ranqueamento** (e não em relação ao valor de *fitness*). Essa abordagem evita que a maior parte da seleção seja feita por indivíduos mais aptos.
- Na seleção por torneio, são escolhidos  $n$  cromossomos de forma aleatória. O indivíduo com **a melhor aptidão dentro desse grupo** é escolhido para gerar descendentes, e esse processo é repetido até que um número suficiente de "pais" para a próxima geração seja selecionado.
- O **elitismo** é uma estratégia de **preservação dos melhores indivíduos**. Como depois da seleção os indivíduos geram descendentes, os melhores cromossomos podem se perder no processo. Usando o elitismo, os melhores cromossomos, além de serem selecionados, **são transferidos para a próxima geração sem sofrer alterações**.

# OPERADORES GENÉTICOS

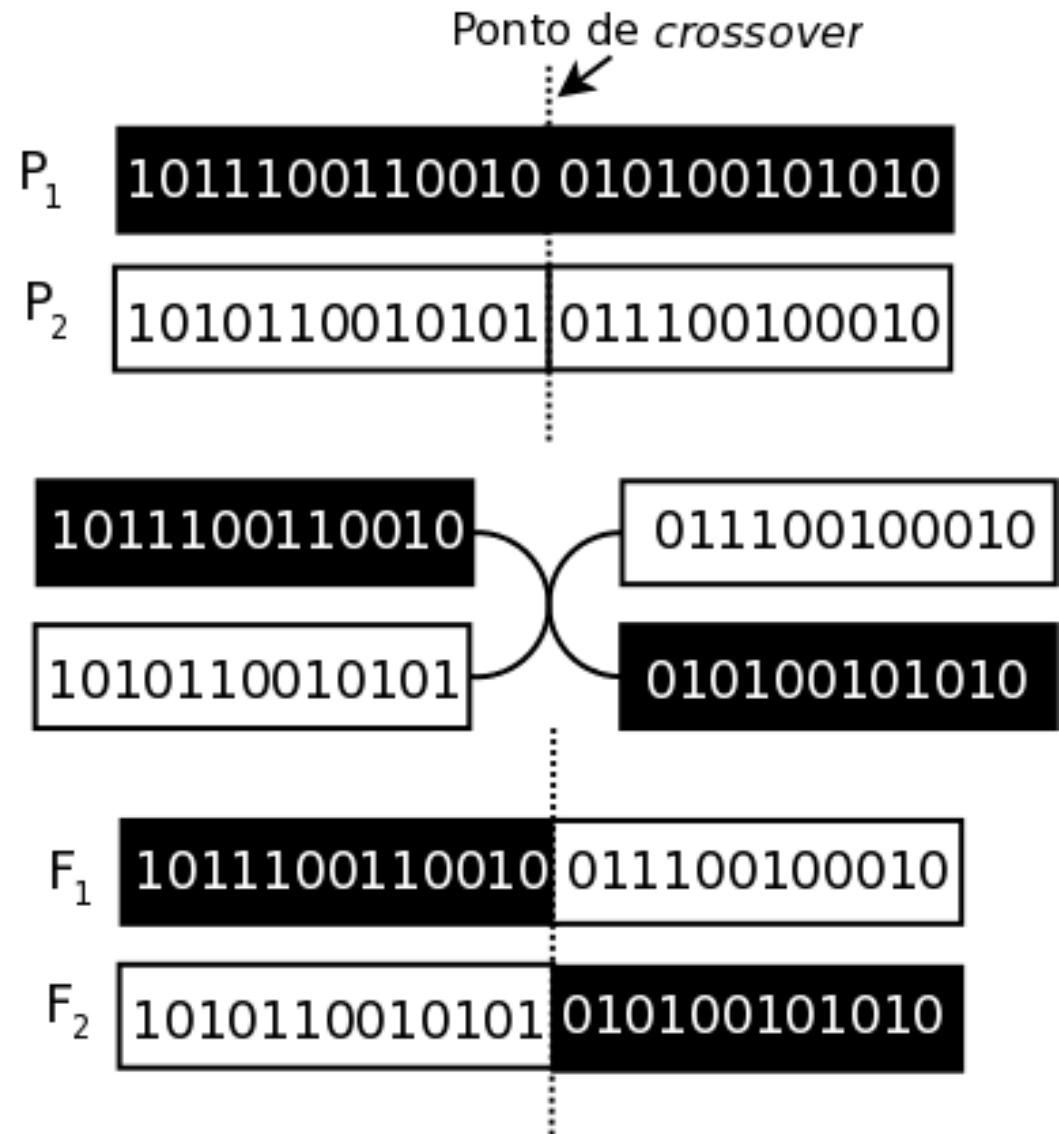
- Os operadores genéticos são responsáveis **por gerar novos indivíduos (nova população) a partir da seleção feita na população inicial.**
- A cada nova geração, novos indivíduos vão surgindo, os quais, em teoria, **são melhores (mais aptos) do que seus pais.**
- Os operadores mais difundidos na literatura de AGs são os operadores de crossover (ou recombinação) e a mutação.
- Assim nos processos biológicos, os fenômenos da recombinação e da mutação, podem ou não ocorrer, havendo certa probabilidade de ocorrência.

# Crossover

- O operador crossover cria novos indivíduos através **da recombinação de partes diferentes de dois cromossomos pais** escolhidos na etapa de seleção. A partir de dois pais são gerados dois descendentes.
- Esse operador é aplicado com uma determinada probabilidade de ocorrência, chamada **taxa de crossover**.
- Existem diferentes operadores crossover, estando muito atrelados a escolha da codificação feita para o problema. Vamos ver a aplicação de dois: o crossover para codificação binária e o para codificação real.

# CROSSOVER PARA CODIFICAÇÃO BINÁRIA

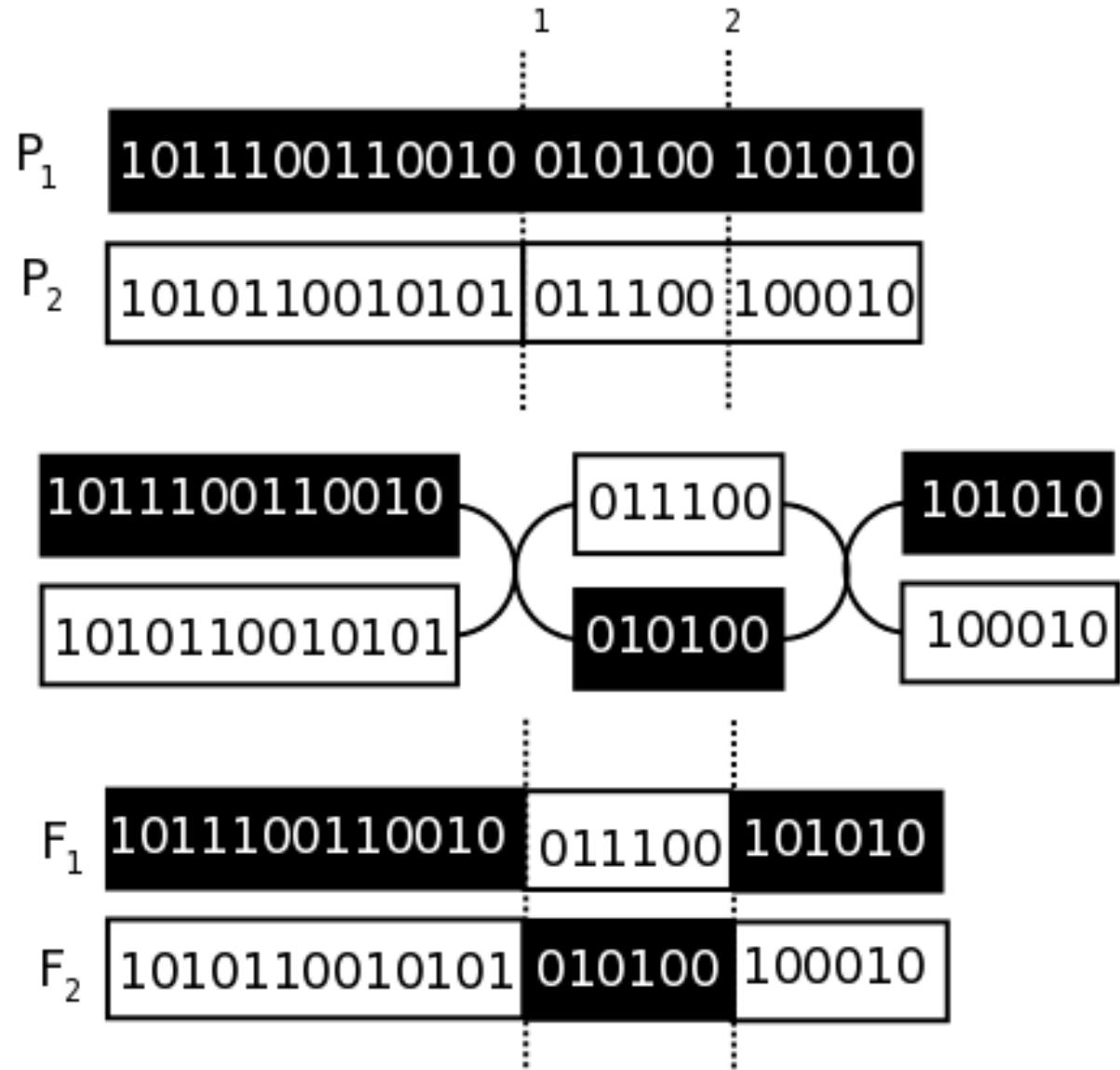
- Para os crossovers binários existem os métodos de um **ponto e dois ou vários pontos**.
- Dados dois cromossomos-pai, P<sub>1</sub> e P<sub>2</sub>, o operador crossover de um ponto faz um “corte” numa posição escolhida aleatoriamente em ambos os cromossomos. A partir das partes geradas com o corte, dois novos indivíduos serão gerados (F<sub>1</sub> e F<sub>2</sub>), concatenando as partes dos cromossomos-pai.



# CROSSOVER PARA CODIFICAÇÃO BINÁRIA

- O operador crossover de vários pontos realiza a mesma operação, extendendo os cortes para os vários pontos definidos.

Pontos de crossover



# CROSSOVER PARA CODIFICAÇÃO REAL

- Os mesmos operadores de crossover binários podem ser aplicados na codificação real. Mas, para além da simples recombinação, outros operadores podem ser usados, como:

- Crossover aritmético: os cromossomos descendentes são gerados a partir de uma combinação linear dos cromossomos-pai ( $P_1$  e  $P_2$ ), utilizando um valor de  $\alpha$  gerado aleatoriamente no intervalo 0 e 1.

$$F_1 = \alpha \times P_1 + (1 - \alpha) \times P_2$$

$$F_2 = \alpha \times P_2 + (1 - \alpha) \times P_1$$

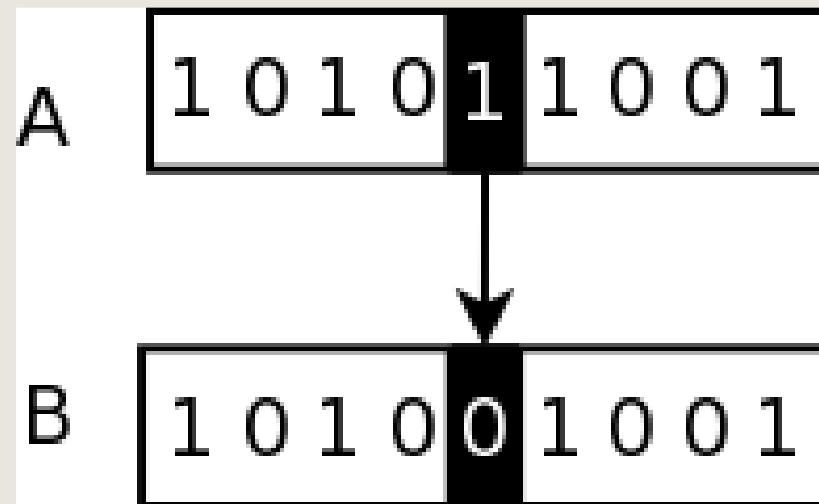
- Crossover heurístico: este operador pode gerar um ou nenhum cromossomo a partir dos pais ( $P_1$  e  $P_2$ ), utilizando a regra:

$$F_1 = r \times (P_2 - P_1) + P_2$$

desde que  $P_2$  tenha um valor de aptidão melhor que de  $P_1$ , e  $r$  é um número aleatório gerado no intervalo 0 e 1.

# MUTAÇÃO

- Os operadores de mutação são responsáveis por explorar de forma global o espaço de busca, **introduzindo material genético novo nos indivíduos já existentes**. A ideia é introduzir uma variabilidade na população, mas sem destruir o progresso que já foi feito.
- Geralmente o valor escolhido para a **taxa de mutação** é bem baixo.
- Na mutação de codificações binárias, **um gene do cromossomo é selecionado aleatoriamente e tem o seu valor invertido**.



# MUTAÇÃO

- Já para a codificação real, aplica-se geralmente as mutações uniforme e Gaussiana:
  - **Mutação uniforme** aplica uma simples substituição de um gene por um número aleatório dentro de um intervalo de valores permitidos para o gene descendente. Esse número aleatório é gerado com base na distribuição uniforme.
  - **Mutação Gaussiana** é uma variação da mutação uniforme que substitui o gene por um número aleatório dentro de uma distribuição normal.

$$c_i = \begin{cases} N(p_i, \sigma), & \text{se } i = j \\ p_i, & \text{caso contrário} \end{cases}$$

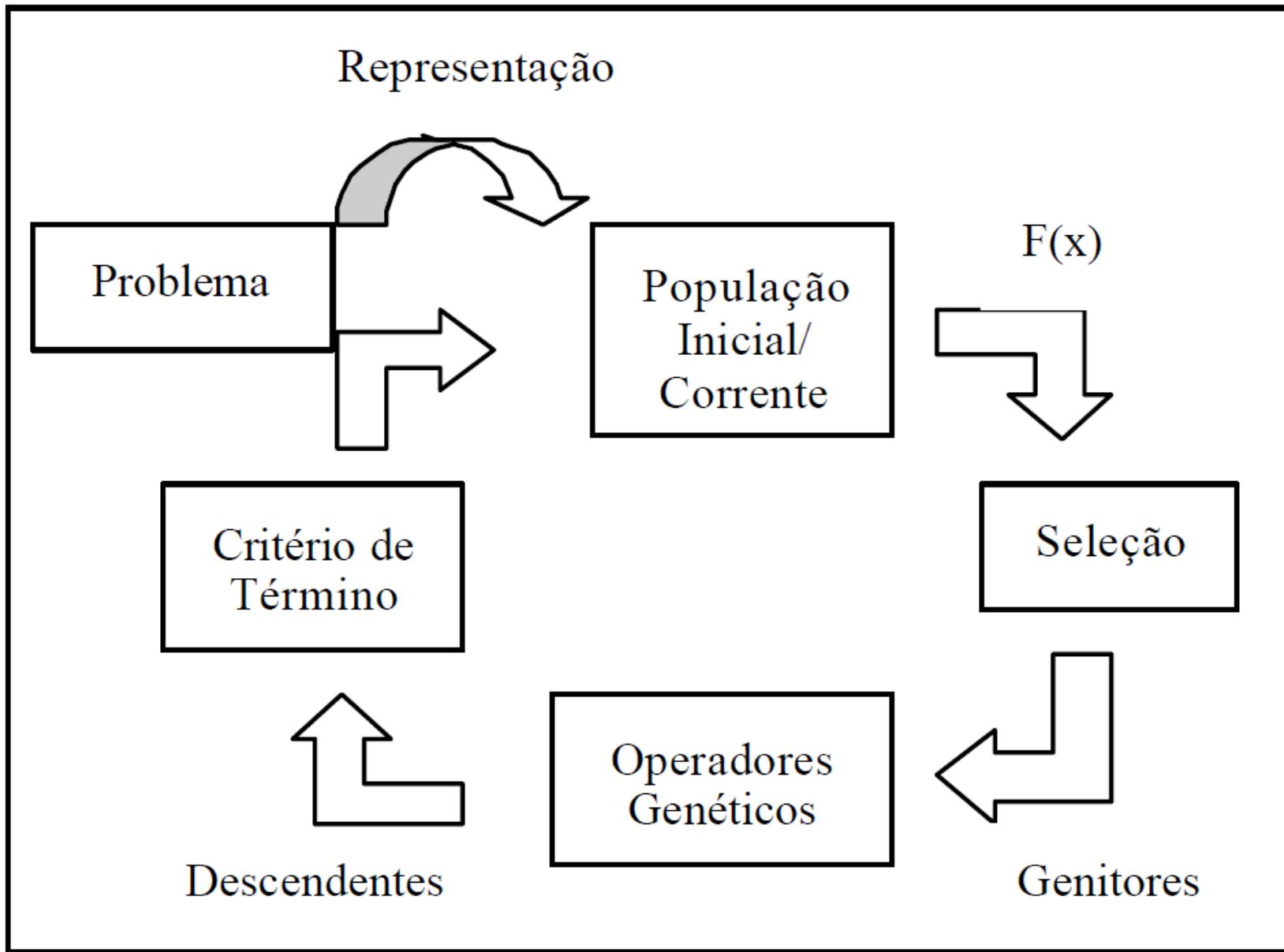
onde  $N(p_i, \sigma)$  é a distribuição normal com média  $p_i$  e desvio padrão  $\sigma$ .

# PARÂMETROS E CRITÉRIOS DE PARADA

- Em AGs, vários parâmetros controlam o processo evolucionário:
  - **Tamanho da população:** número de pontos do espaço de busca sendo considerados em paralelo a cada ciclo.
  - **Taxa de crossover:** probabilidade de um indivíduo ser recombinado com outro.
  - **Taxa de mutação:** probabilidade do conteúdo de um gene ser alterado.
  - **Número de gerações:** total de ciclos de evolução de um AG.
  - **Total de indivíduos:** total de tentativas em um experimento (tamanho da população x número de gerações).

Os dois últimos parâmetros são em geral empregados como critério de parada de um algoritmo genético.

# FUNCIONAMENTO DE UM AG



# EXEMPLO

Vamos maximizar uma função multimodal

$$f(x) = (x \bmod 7)^2 + \lfloor x/3 \rfloor$$

$$x \in \{0, 1, 2, \dots, 31\}$$

Utilizando uma codificação binária e seleção por torneio.

