

# Programação Genética

Airton Bordin Junior  
[airtonbjunior@gmail.com]

Mestrado em Ciência da Computação – Inteligência Computacional  
Profº Dr Celso Gonçalves Camilo Junior

Universidade Federal de Goiania (UFG) - Instituto de Informática - Maio/2017

# Programação

- Introdução
- Heurísticas e Metaheurísticas
- Algoritmos evolucionários
- Programação Genética
- Referências





# Introdução

- Podemos classificar os problemas computacionais em
  - Tratáveis, também chamados de polinomiais
    - Podem ser resolvidos por algoritmos determinísticos.
  - Intratáveis, ou não polinomiais
    - Sem algoritmo determinístico para resolver o problema em tempo hábil.

Tratáveis

- Polinomiais
- Algoritmos determinísticos

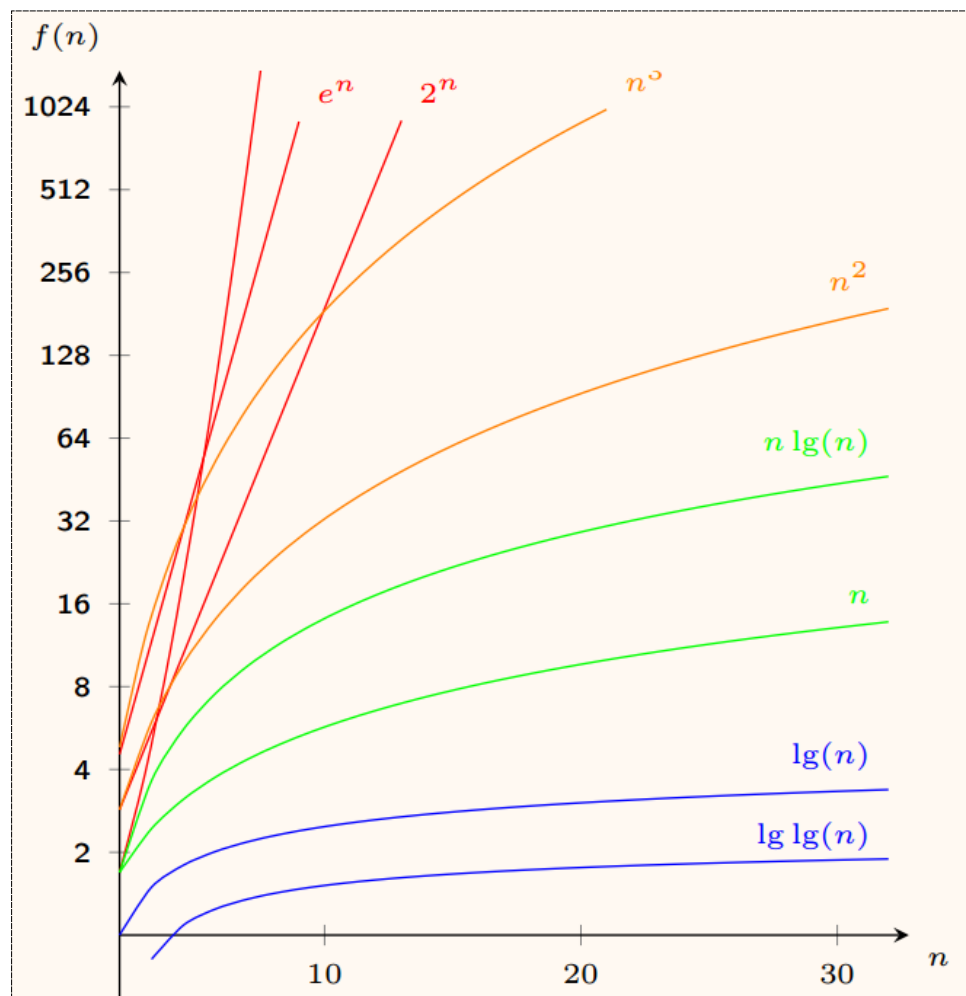
Intratáveis

- Não polinomiais
- Algoritmos não determinísticos
- Solução determinística inviável

# Introdução

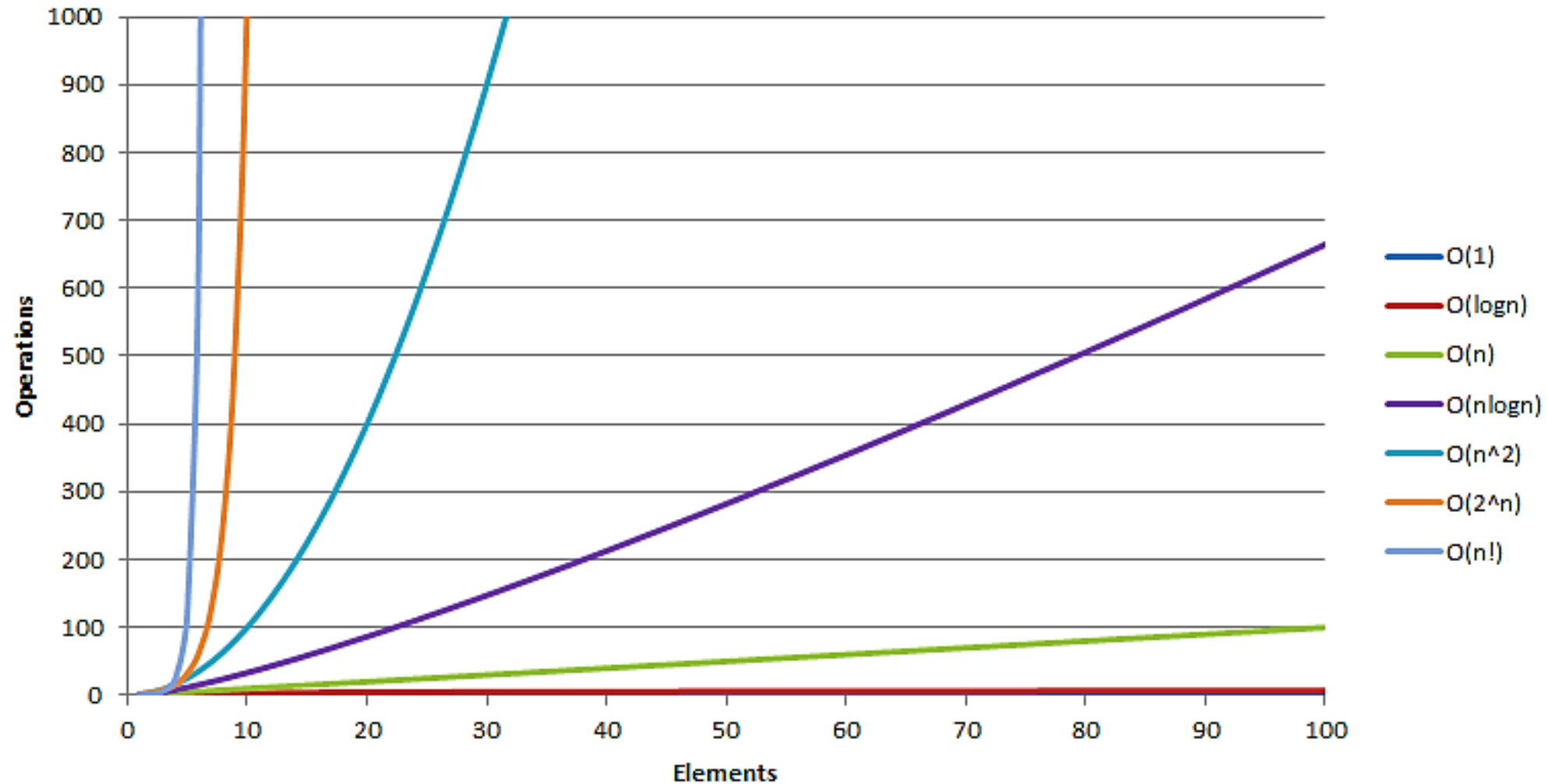


[CAPPELLE, 2017]





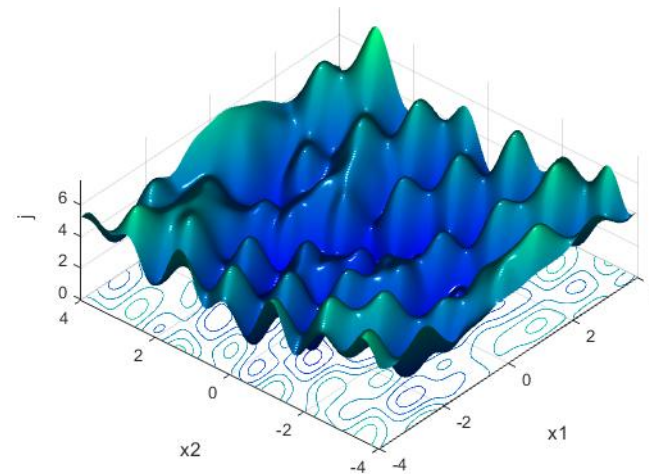
# Introdução





# Heurística

- Impraticabilidade de encontrar/calcular a melhor resposta para problemas não polinomiais;
- Desafio: produzir, em tempo reduzido, soluções tão próximas quanto possíveis da solução ótima.

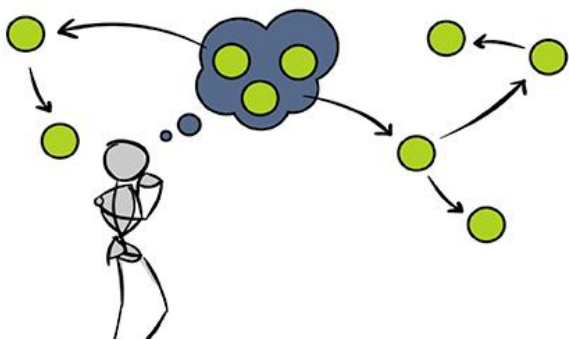




# Metaheurística

## Propriedades e características das metaheurísticas

[SALIBA, 2010]



Estratégias que guiam o processo de busca;

Exploração eficiente do espaço de busca - soluções ótimas ou quase ótimas;

De simples procedimentos de busca local a complexos processos de aprendizado;

Aproximados e usualmente não determinísticos;

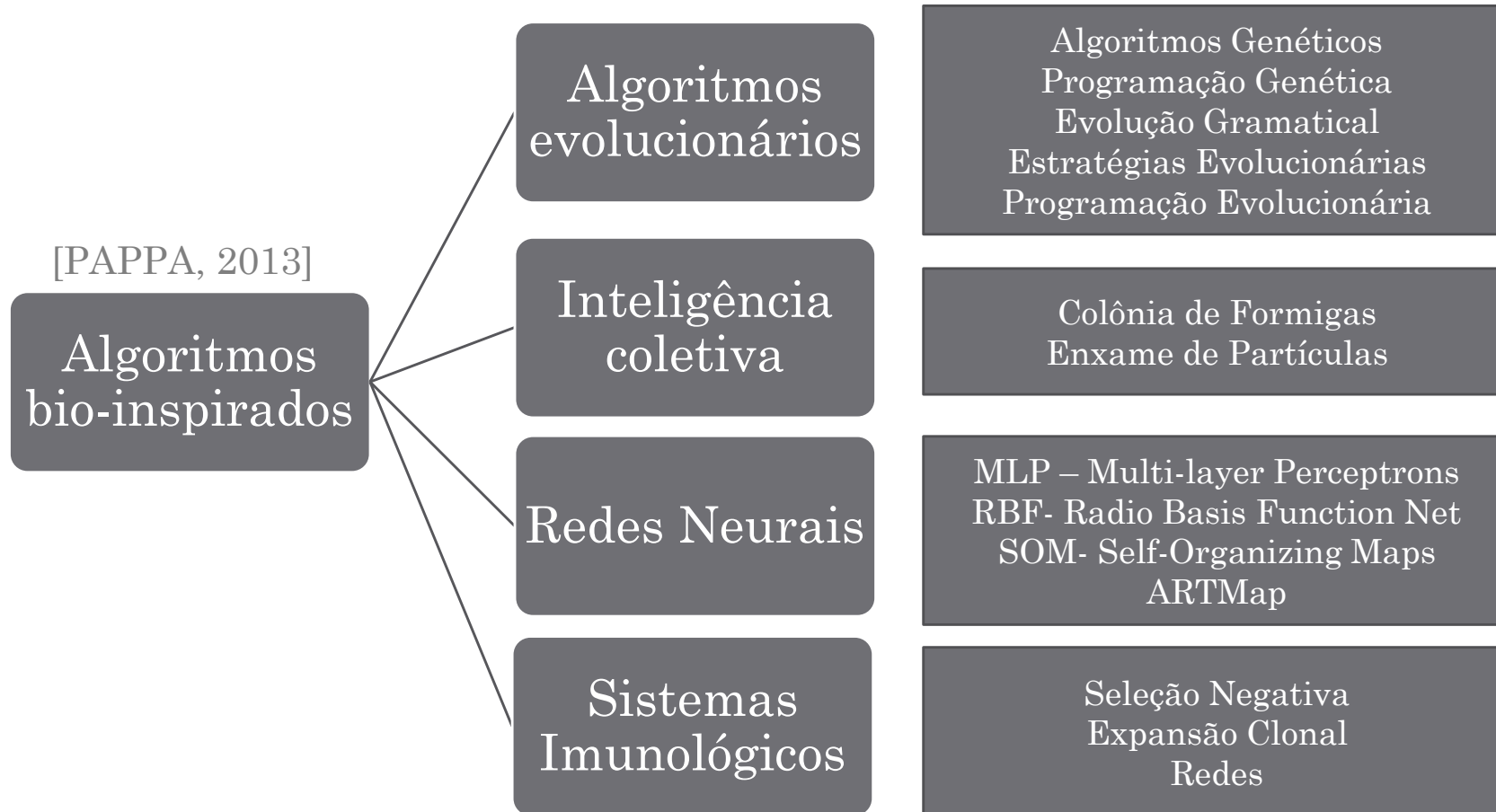
Podem incorporar mecanismos para evitar ficar presos em áreas confinadas do espaço de busca;

Não são específicas para um determinado problema;

Podem usar um conhecimento específico do problema na forma de heurísticas que são controladas por uma estratégia de nível superior.



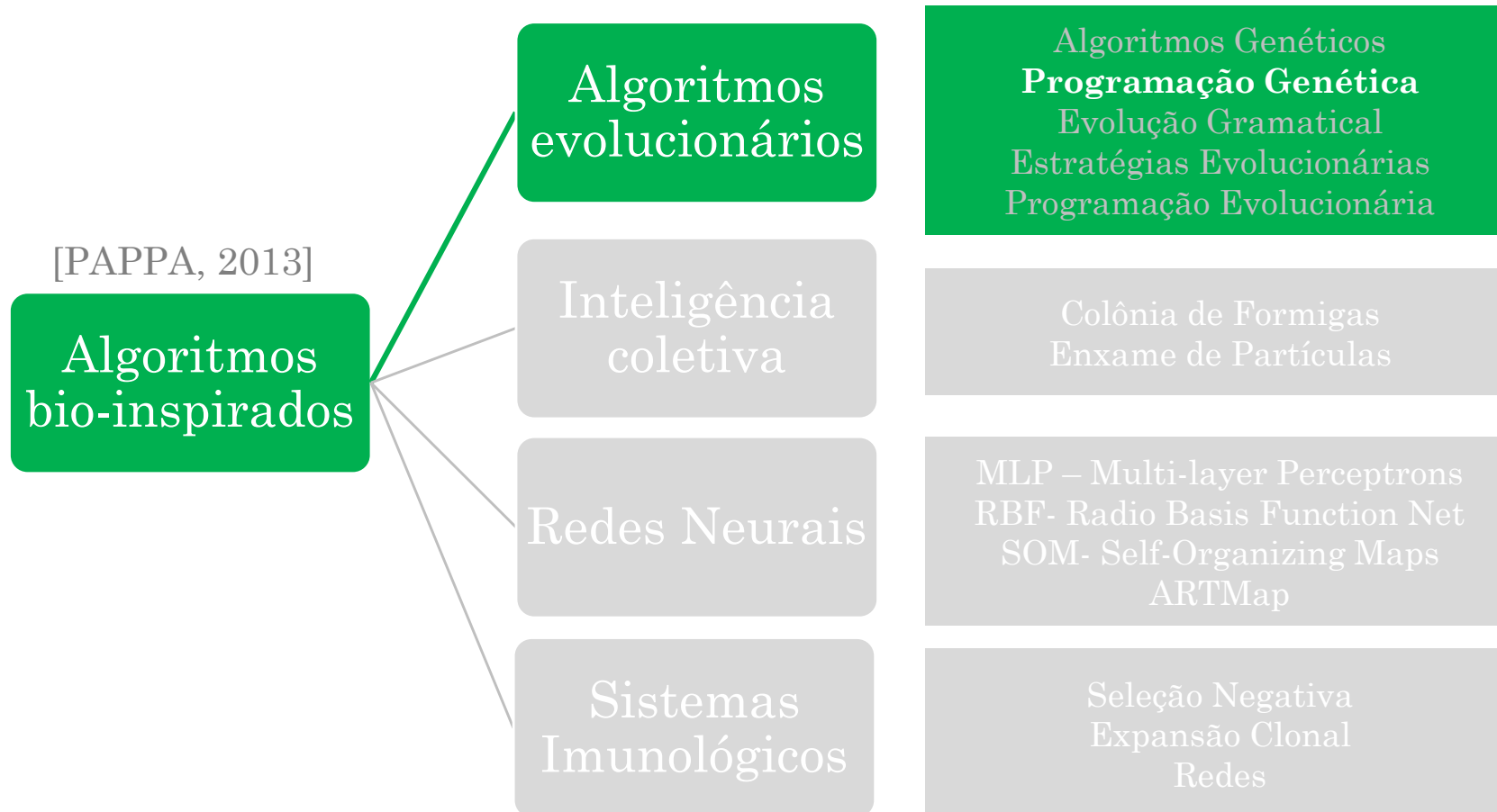
# Algoritmos bio-inspirados







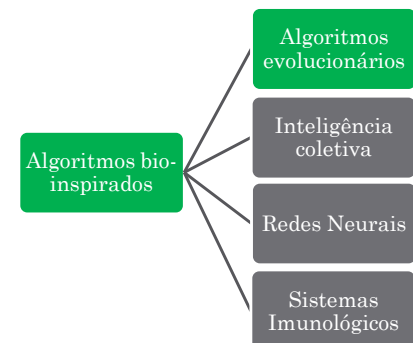
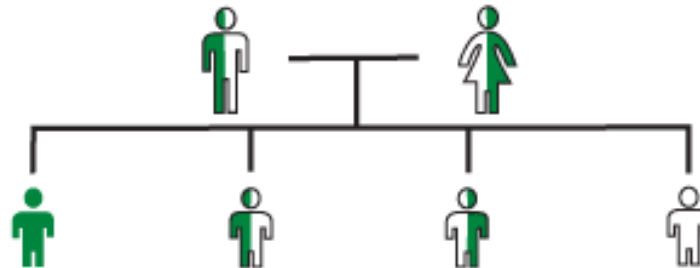
# Algoritmos bio-inspirados





# Algoritmos evolucionários

- Inspirados na teoria de evolução de Darwin;
- Evolução: mudança das características (genéticas) de uma população de uma geração para a próxima
  - Mutação dos genes;
  - Recombinação dos genes dos pais.

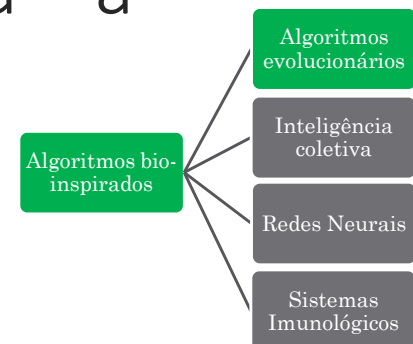




# Algoritmos evolucionários

- Evolução é caracterizada basicamente por um processo constituído de 3 passos [VON ZUBEN, 2005]

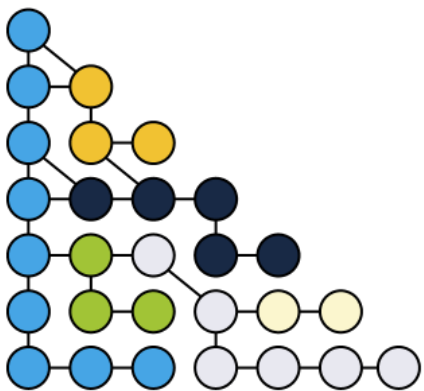
1. Reprodução com herança genética;
2. Introdução de variação aleatória em uma população de indivíduos;
3. Aplicação da “seleção natural” para a produção da próxima geração.





# Algoritmos genéticos - Revisão

Principais  
diferenças com  
métodos  
tradicionais



Trabalham com uma codificação do conjunto de parâmetros e não com os próprios parâmetros;

Trabalham com uma população e não com um único ponto;

Utilizam informações de custo ou recompensa e não derivadas ou outro conhecimento auxiliar;

Utilizam regras de transição probabilísticas e não determinísticas.



# Algoritmos genéticos - Revisão

## Algumas características

**Paralelo:** mantém uma população de soluções que podem ser avaliadas simultaneamente;

**Global:** AGs não usam somente informações locais, logo não necessariamente ficam presos em máximos locais;

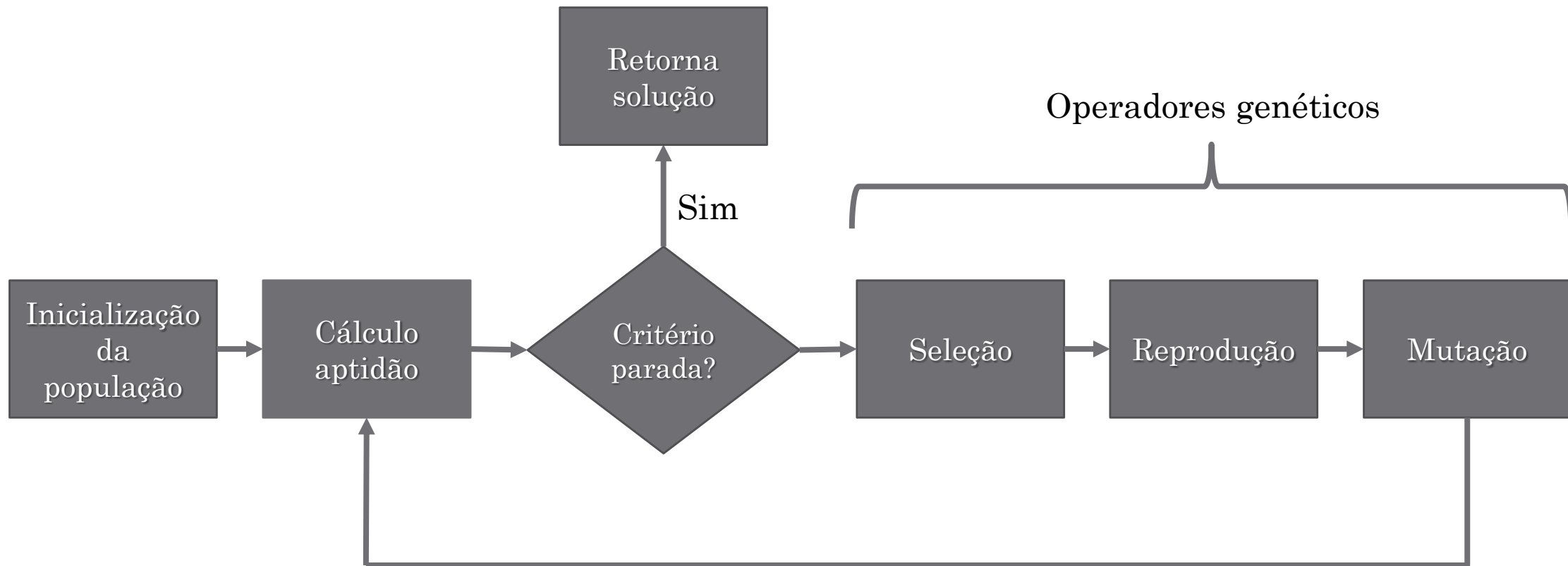
**Não totalmente aleatório:** usam informações da população atual para determinar o próximo estado de busca;

**Não afetados por descontinuidades:** não usam informações de derivadas nem necessitam informações de seu entorno;

**Funções:** Lidam com funções discretas e contínuas.



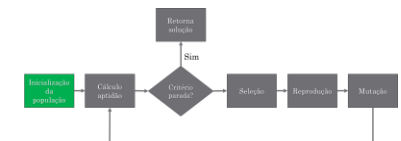
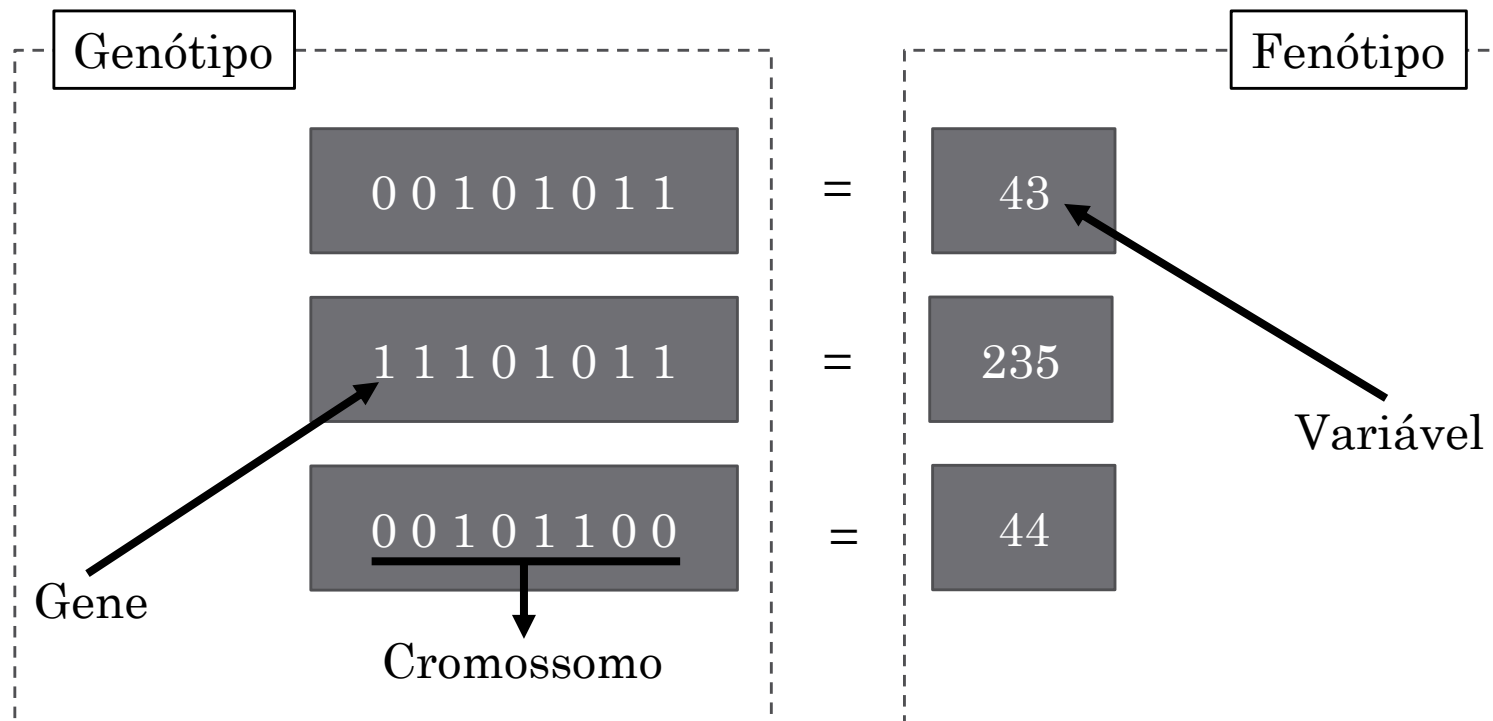
# Algoritmos genéticos - Revisão





# Inicialização da população - Revisão

- Exemplo: representação em bits



# Programação Genética





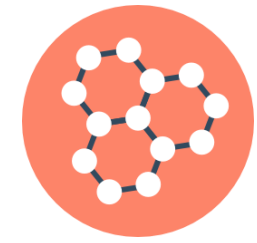
# Programação genética

*Como computadores podem resolver problemas sem serem explicitamente programados para tal?*



# Programação genética

- *Como computadores podem resolver problemas sem serem explicitamente programados para tal?*
- Evolução de programas computacionais
  - Analogias com mecanismos utilizados da evolução biológica natural;
- Criação (automatizada) de um programa que resolve um determinado problema.



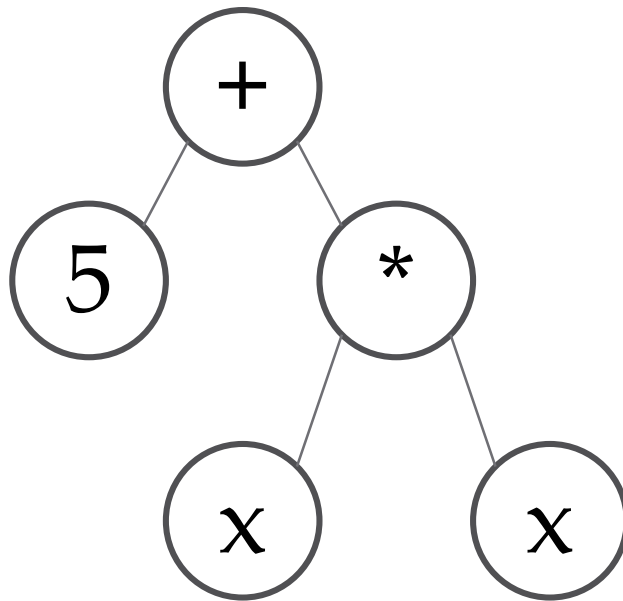
# Programação genética

- *Como computadores podem resolver problemas sem serem explicitamente programados para tal?*
  - Pode ser vista como uma extensão dos Ags
    - Indivíduos são programas;
    - Espaço de busca são todos os possíveis programas.



# Programação genética

- Programas?
  - Funções matemáticas, por exemplo;
  - Representação feita por meio de árvores.



Exemplo programa:  
 $x^2 + 5$

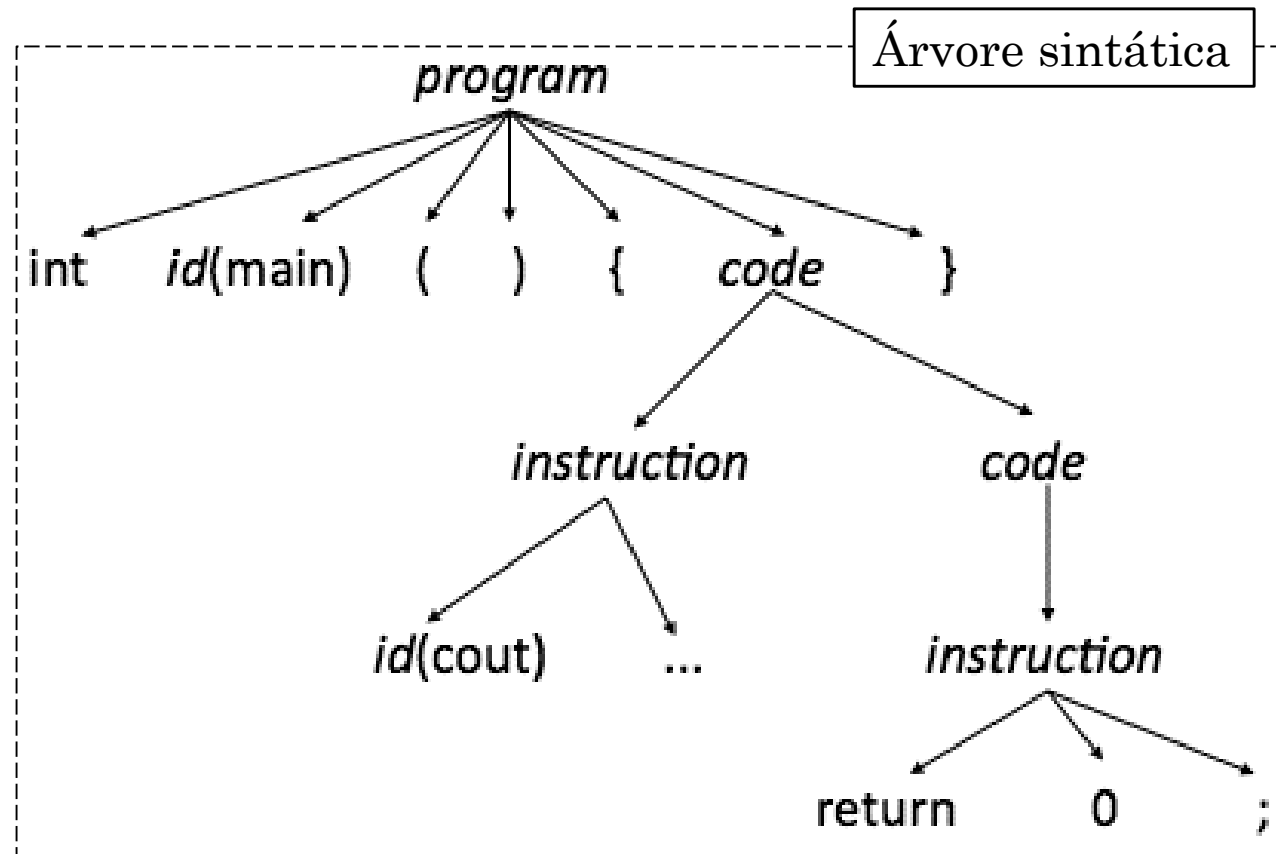


# Programação genética

- Intimamente ligada à ideia de programação funcional (sequência de aplicação de funções a argumentos)
  - Independentemente da linguagem, todos os programas podem ser vistos como uma sequência de aplicações de funções a argumentos;
  - Compiladores usam esse fato para traduzir um programa em uma árvore sintática.



# Programação genética





# Programação genética

- Passos para o correto funcionamento [KOZA, 1992]
  1. Determinar conjunto de terminais;
  2. Determinar conjunto de funções;
  3. Determinar função *fitness*;
  4. Determinar parâmetros e variáveis para controle da execução;
  5. Determinar critério de parada.



# Programação genética

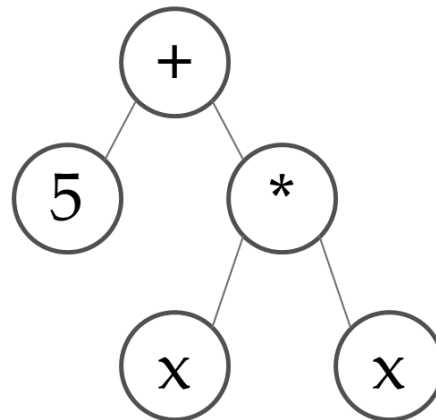
- Criação da população
  - Cria uma população de forma randômica;
  - Profundidade máxima definida por parâmetro;
- Principais métodos
  - Full;
  - Grow;
  - Ramped half-and-half.





# Programação genética

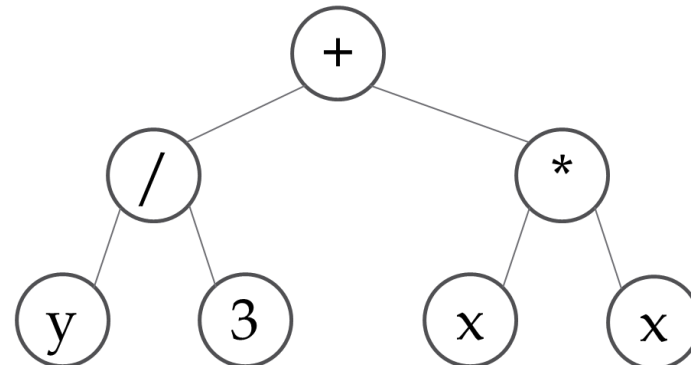
- Criação da população
  - Método Grow
    - Respeita o critério de profundidade máxima da árvore;
    - Escolhe aleatoriamente entre funções e terminais em qualquer nível da árvore, podendo criar estruturas irregulares.





# Programação genética

- Criação da população
  - Método Full
    - Árvores com a profundidade máxima;
    - Escolhe aleatoriamente somente funções, até que um nó de profundidade máxima seja atingido, aí então escolhendo somente terminais.





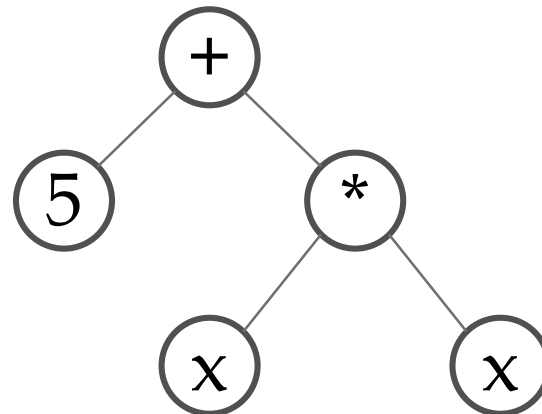
# Programação genética

- Criação da população
  - Método Ramped half-and-half
    - Utiliza o método Grow e Full;
    - Gera um número igual de árvores para cada profundidade;
    - 50% utilizará o método full e 50% o método Grow.



# Programação genética

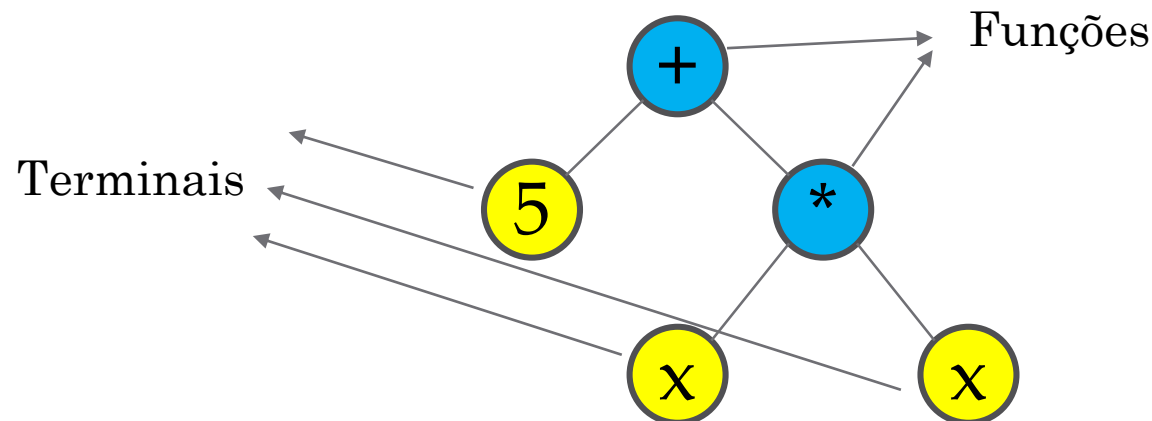
- Criação da população
  - Funções e terminais
    - Funções: funções aritméticas (+, -, /, \*), funções booleanas, funções matemáticas, etc.
    - Terminais: constantes numéricas, dados externos, variáveis.





# Programação genética

- Criação da população
  - Funções e terminais
    - Funções: funções aritméticas (+, -, /, \*), funções booleanas, funções matemáticas, etc.
    - Terminais: constantes numéricas, dados externos, variáveis.





# Programação genética

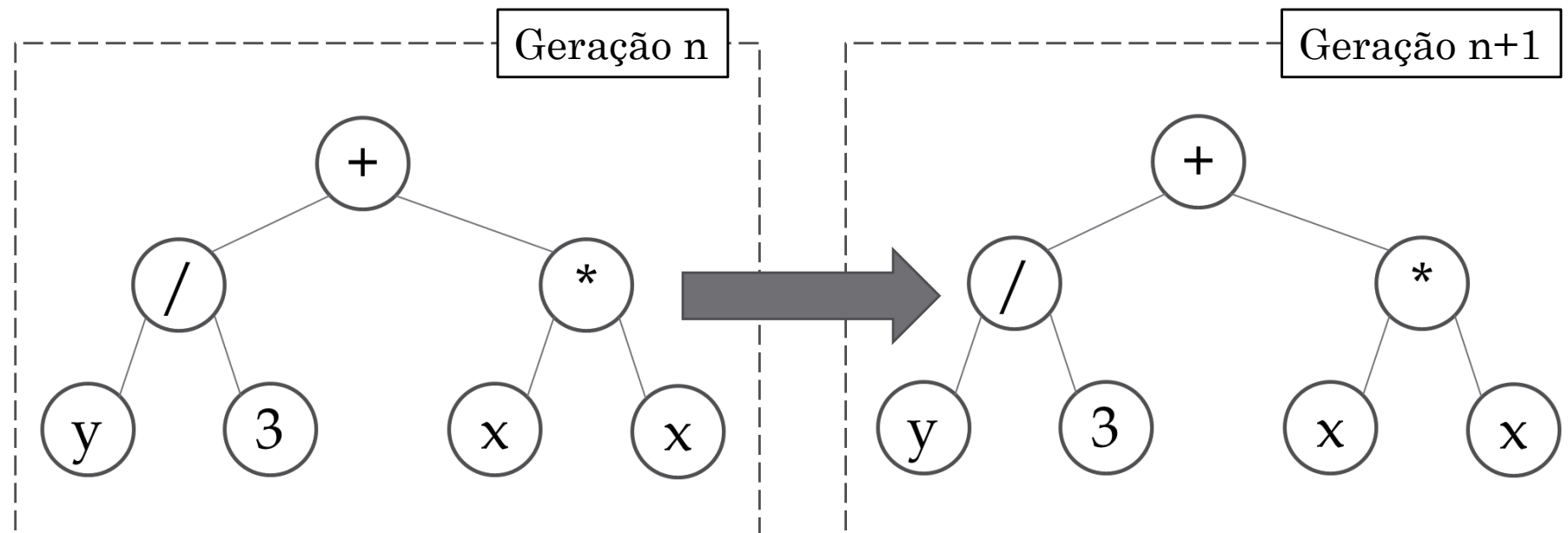
- Operadores genéticos
  - Reprodução;
  - Crossover;
  - Mutação;
  - Permutação;
  - Edição;
  - Encapsulamento;
  - Destruição.



# Operadores genéticos

- Reprodução

- Um indivíduo com uma bom valor após função de avaliação (*fitness*) é escolhido;
- É feita uma cópida idêntica do indivíduo para a próxima geração.

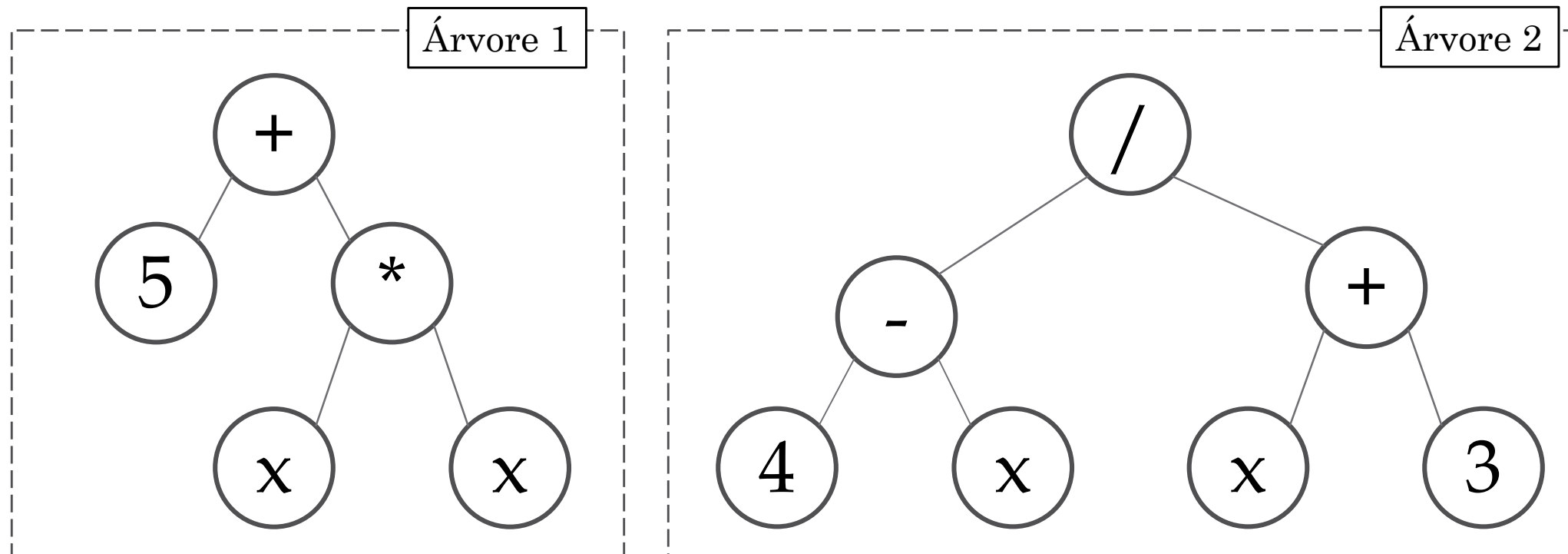




# Operadores genéticos

- Crossover

- Troca entre partes dos indivíduos selecionados;
- Partes escolhidas de forma aleatória nas duas árvores.



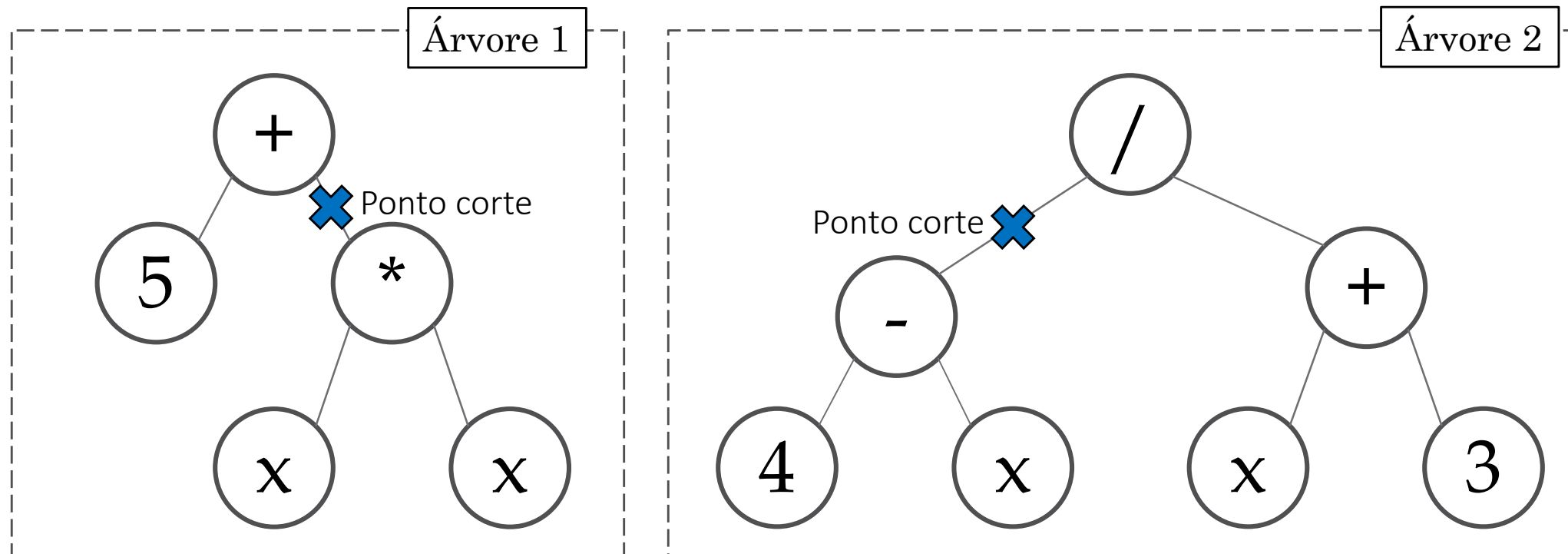




# Operadores genéticos

- Crossover

- Troca entre partes dos indivíduos selecionados;
- Partes escolhidas de forma aleatória nas duas árvores.

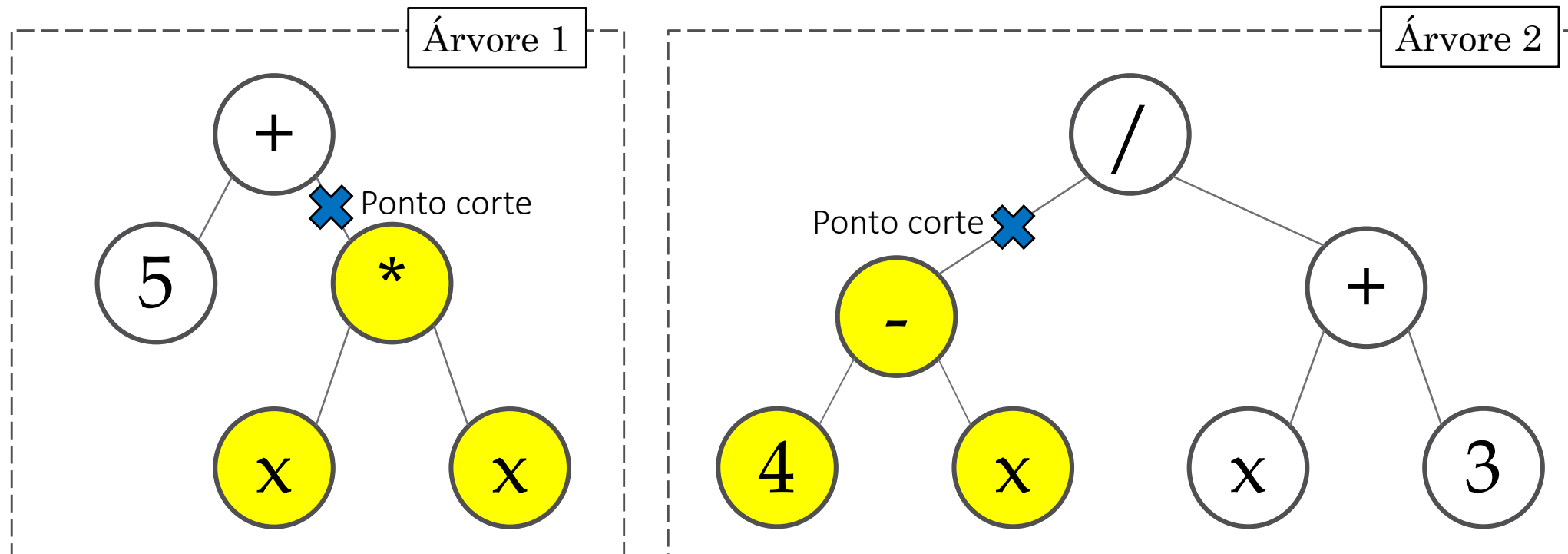


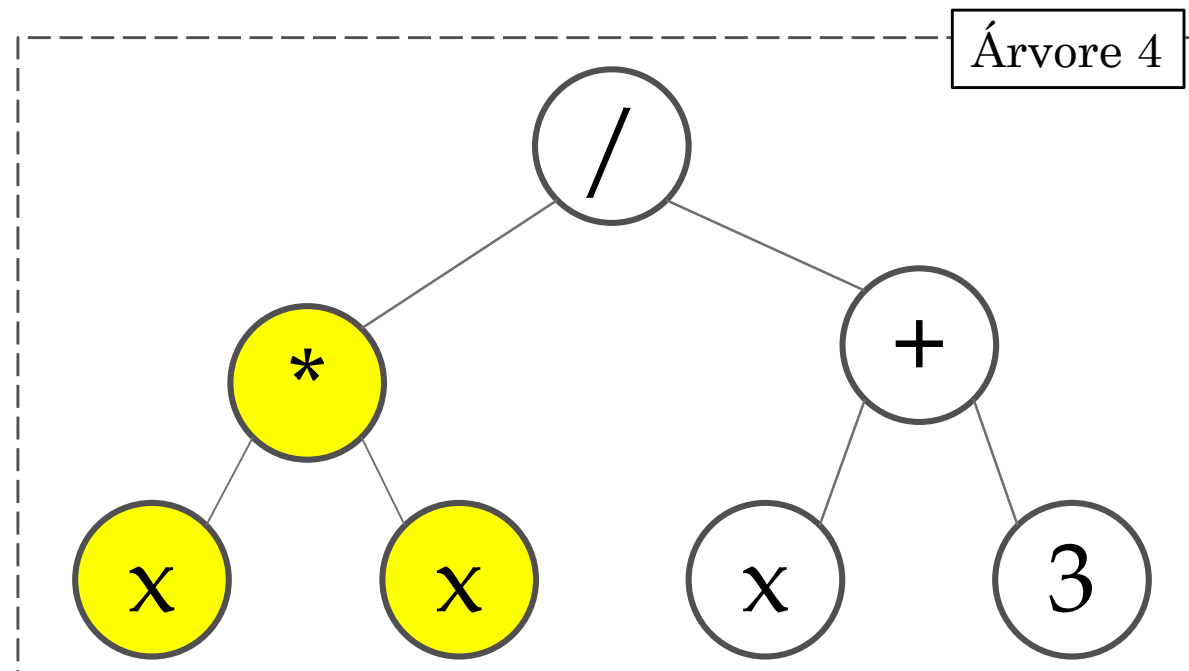
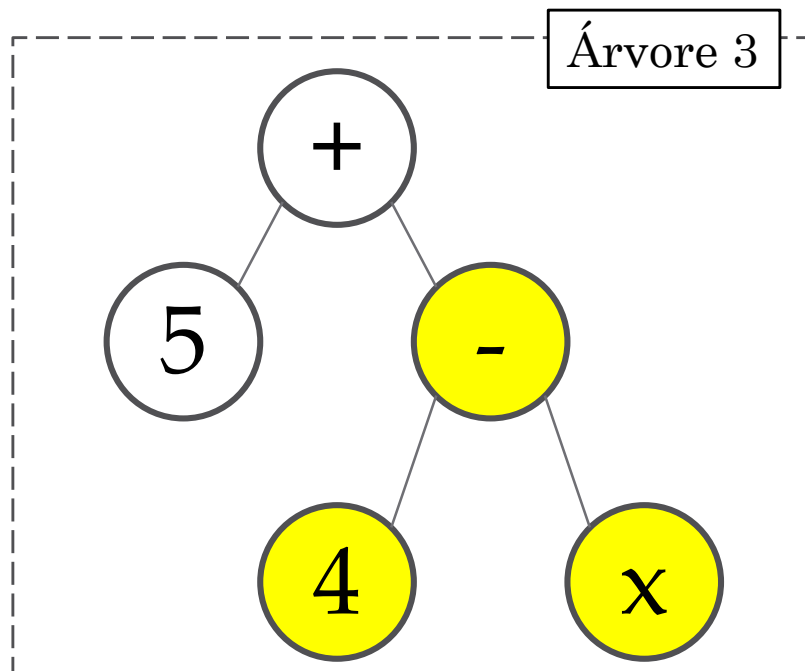
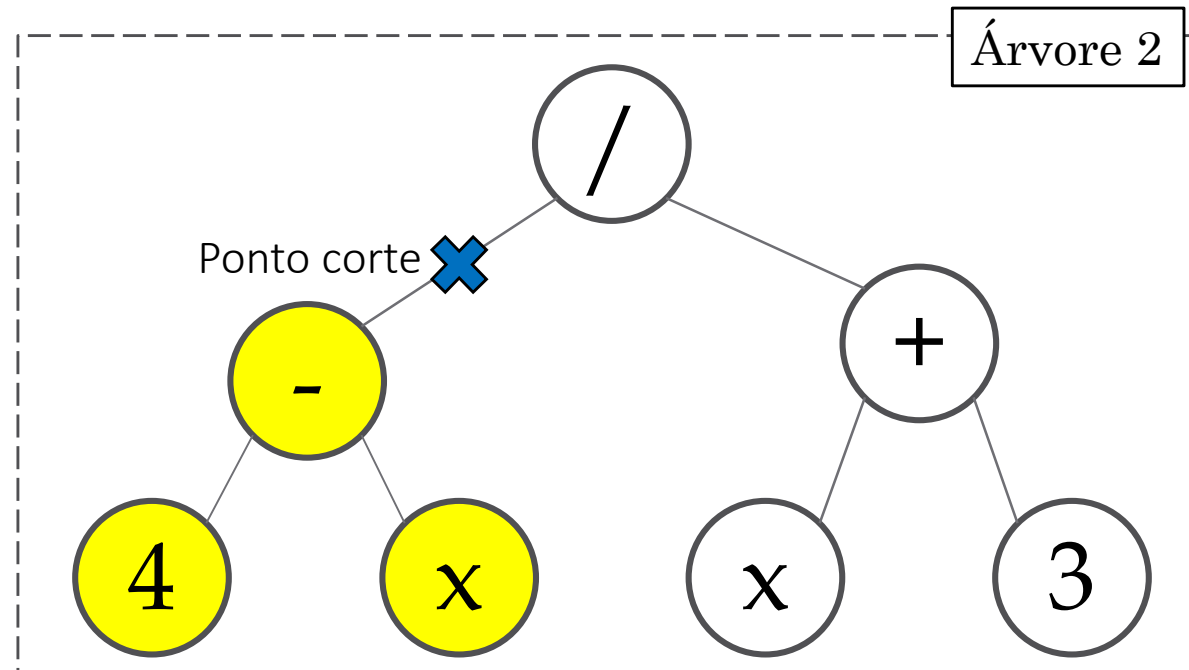
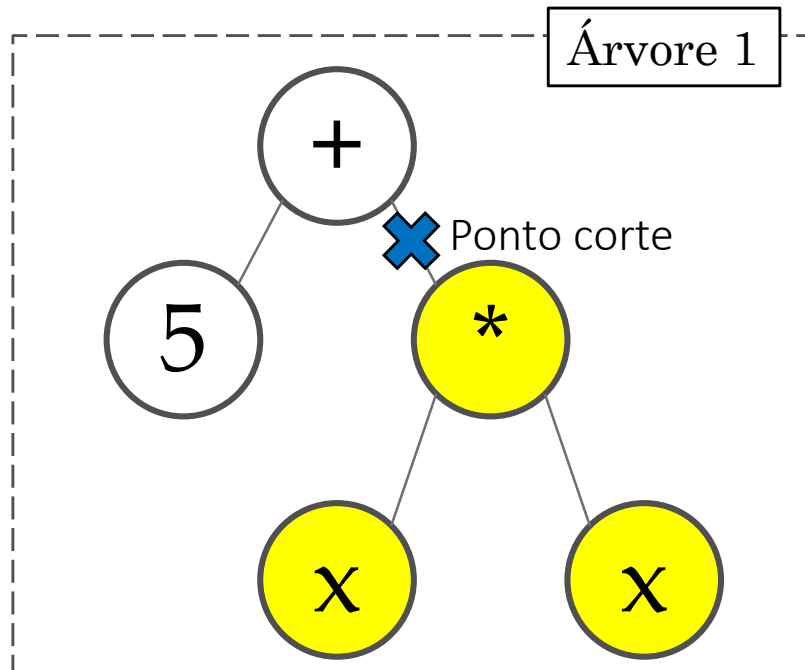


# Operadores genéticos

- Crossover

- Troca entre partes dos indivíduos selecionados;
- Partes escolhidas de forma aleatória nas duas árvores.



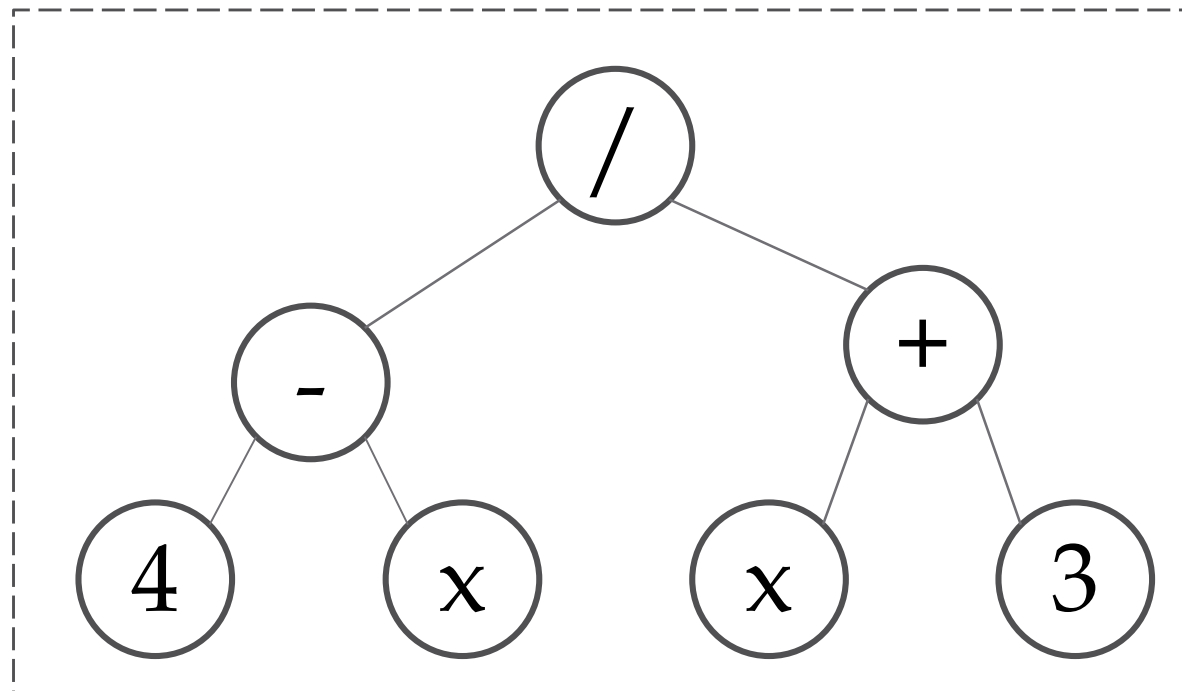




# Operadores genéticos

- Mutaç o

- Mudan a aleat ria em um dos n s da  rvore;
- Adiciona diversidade na popula o.

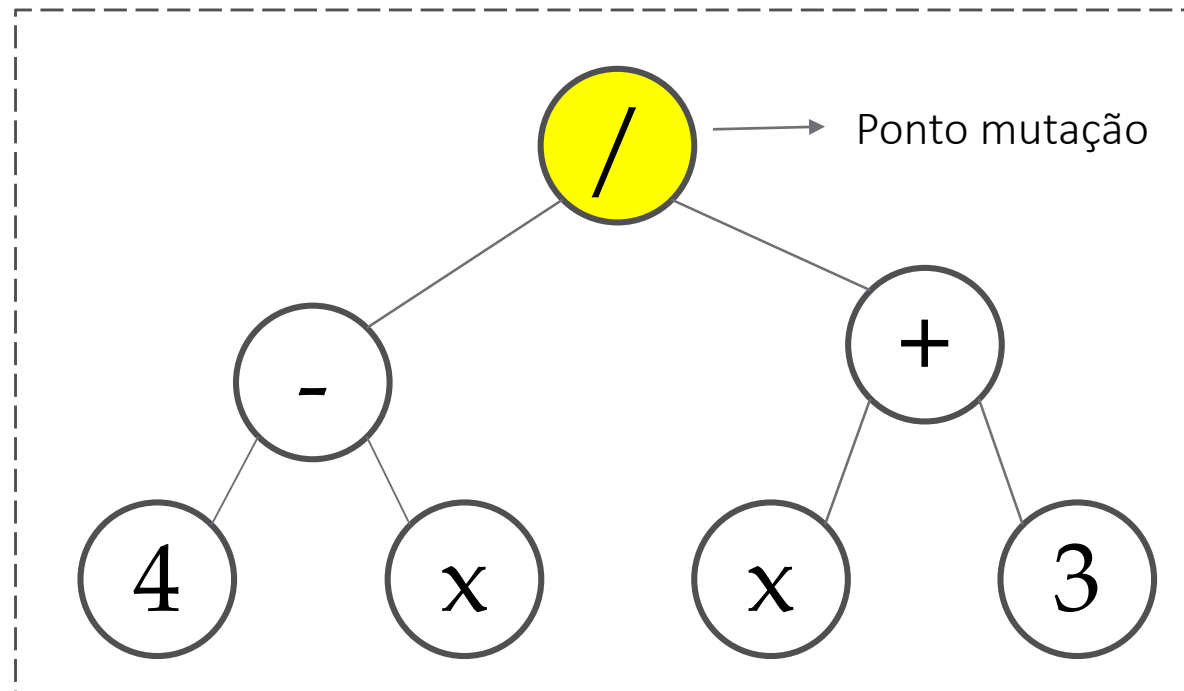




# Operadores genéticos

- Mutaç o

- Mudan a aleat ria em um dos n s da  rvore;
- Adiciona diversidade na popula o.

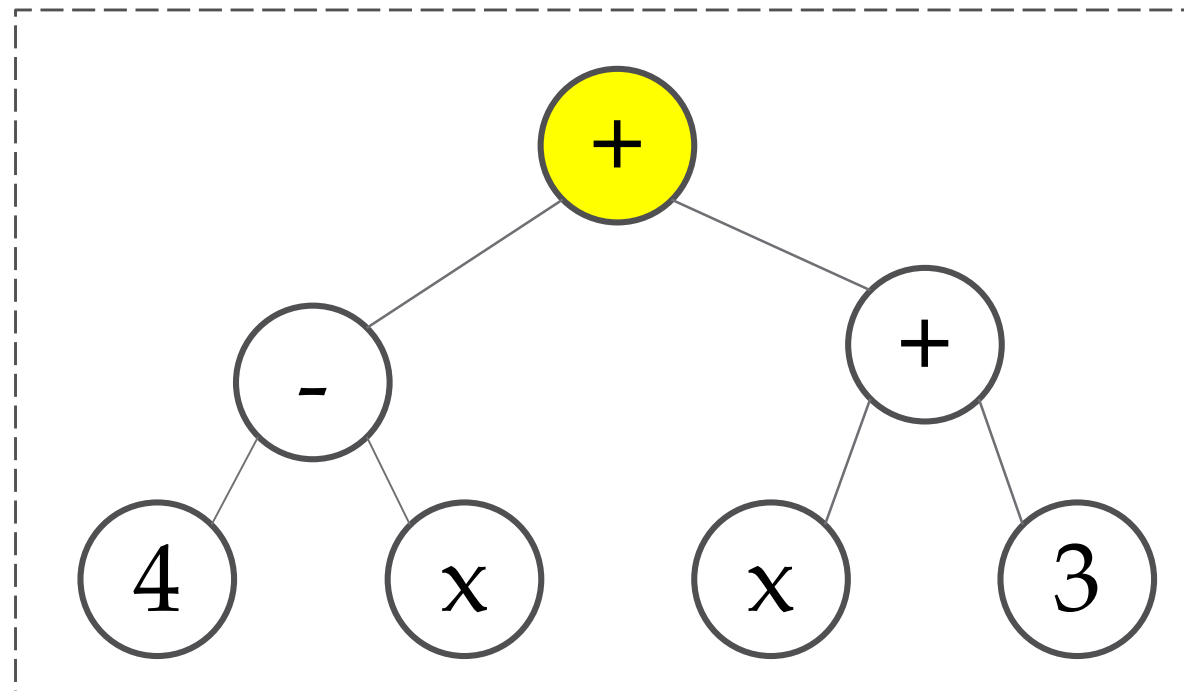




# Operadores genéticos

- Mutaç o

- Mudan a aleat ria em um dos n s da  rvore;
- Adiciona diversidade na popula o.

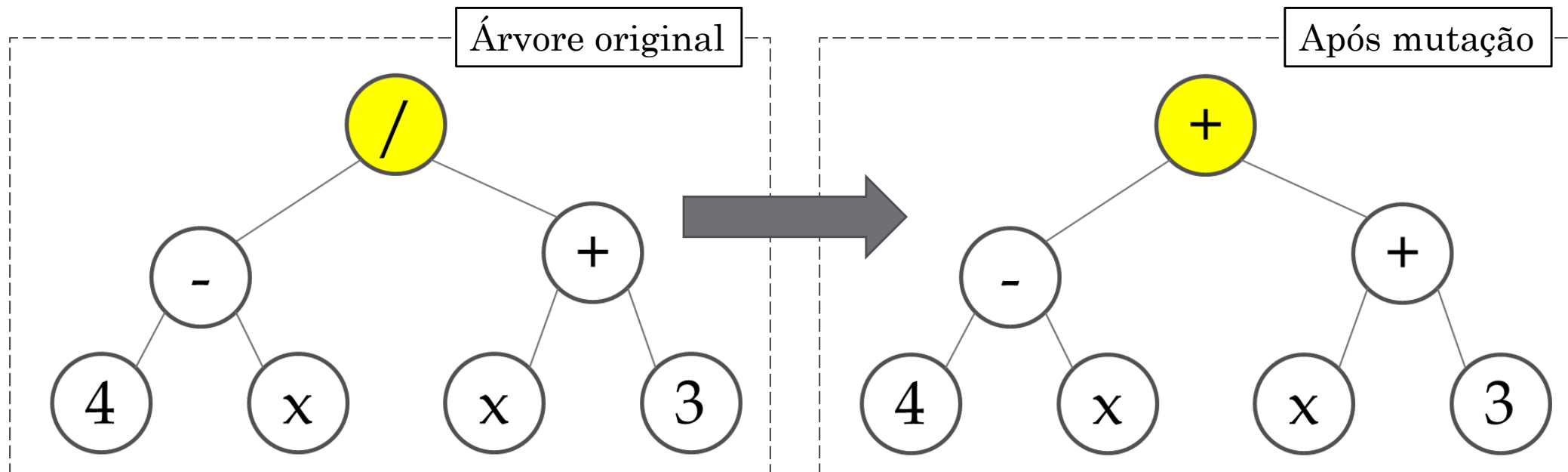




# Operadores genéticos

- Mutaç o

- Mudan a aleat ria em um dos n os da  rvore;
- Adiciona diversidade na popula  o.

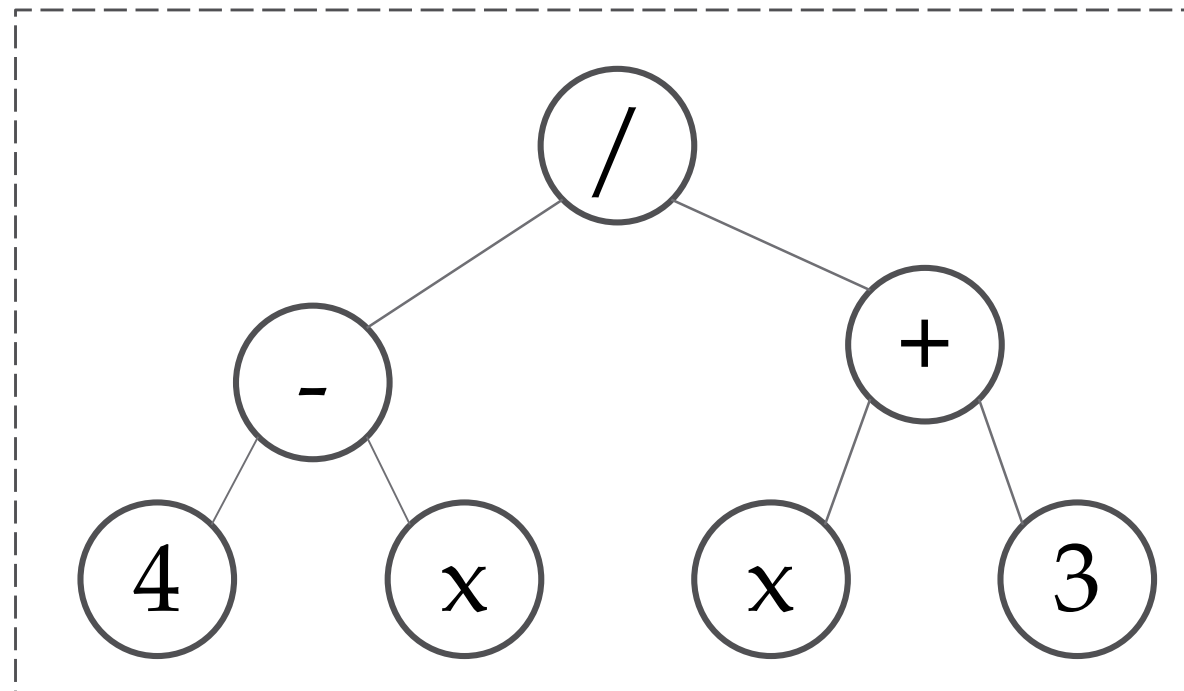




# Operadores genéticos

- Permutação

- Escolhe um ponto aleatório e inverte os terminais e/ou funções.



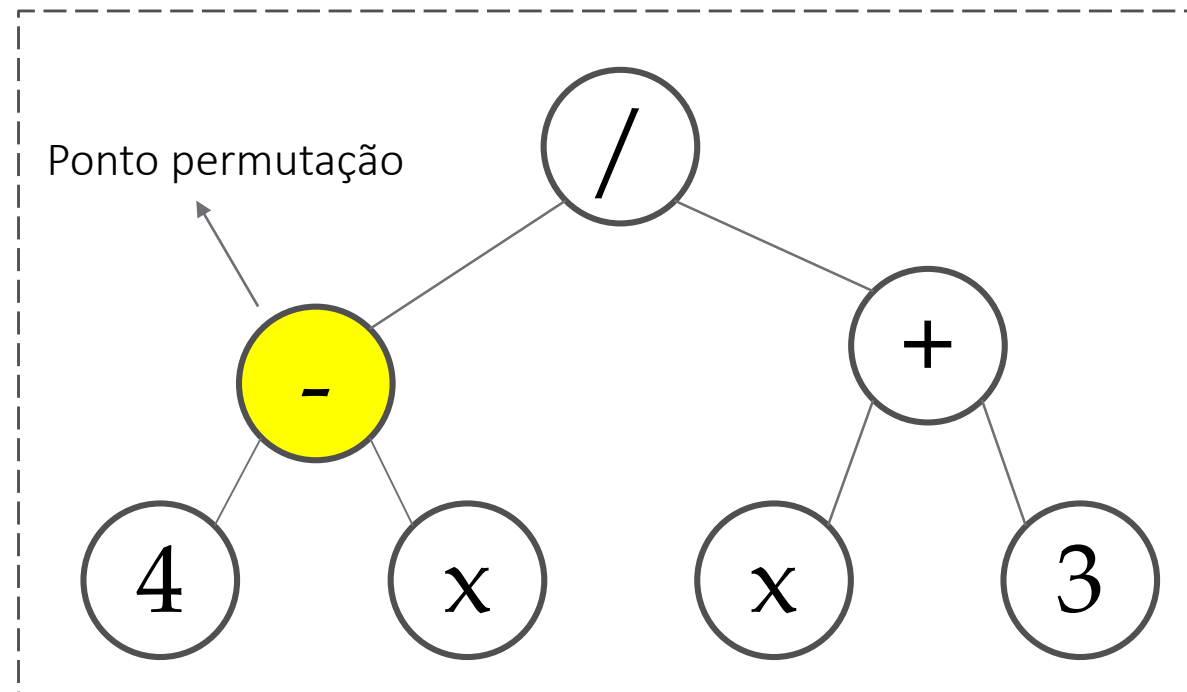




# Operadores genéticos

- Permutação

- Escolhe um ponto aleatório e inverte os terminais e/ou funções.

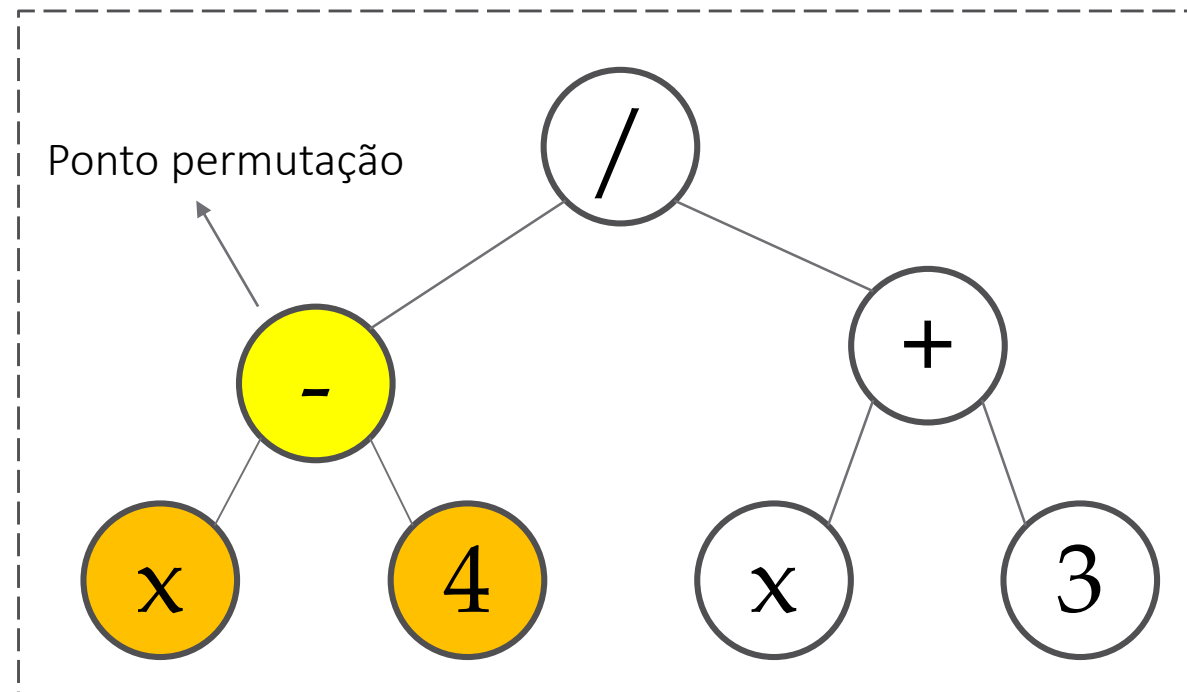




# Operadores genéticos

- Permutação

- Escolhe um ponto aleatório e inverte os terminais e/ou funções.

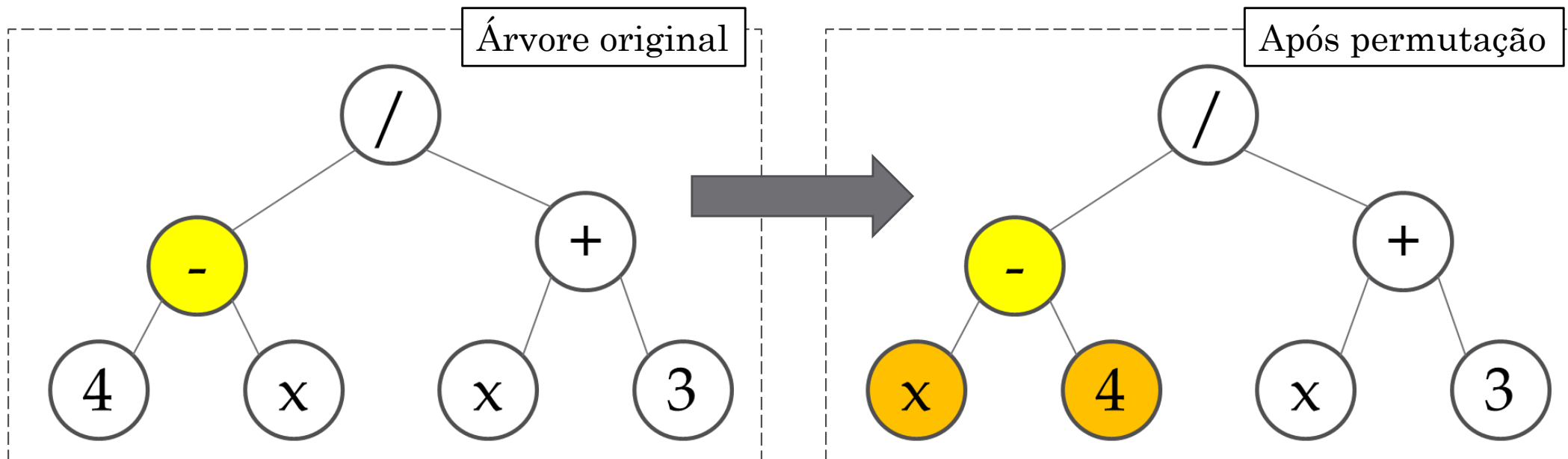


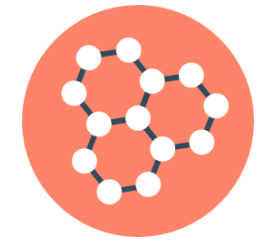


# Operadores genéticos

- Permutação

- Escolhe um ponto aleatório e inverte os terminais e/ou funções.

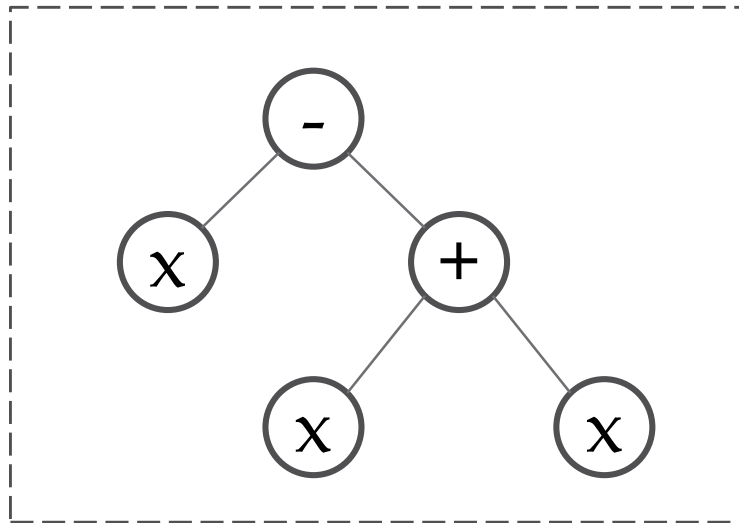




# Operadores genéticos

- Edição

- Forma de simplificação e edição de expressões;
- Muito custosa – Consumo considerável de tempo;
- Torna a expressão menos vulnerável ao crossover.



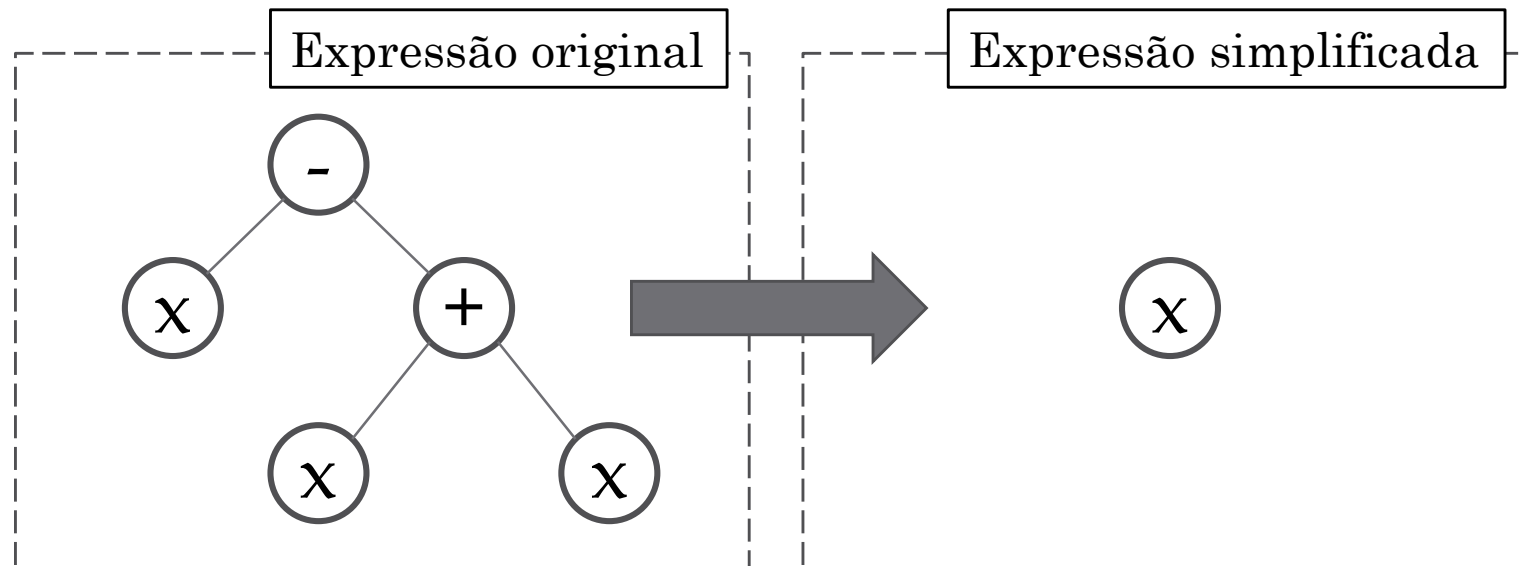
Expressão:  
 $X+X-X$



# Operadores genéticos

- Edição

- Forma de simplificação e edição de expressões;
- Muito custosa – Consumo considerável de tempo;
- Torna a expressão menos vulnerável ao crossover.

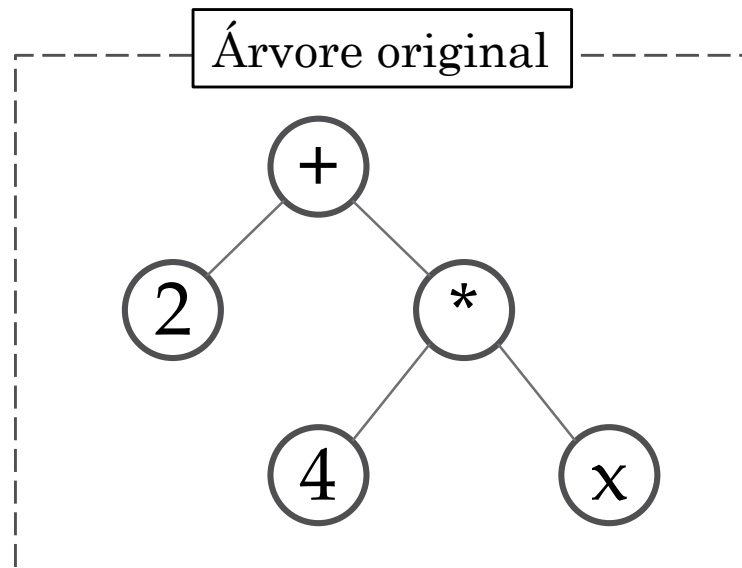




# Operadores genéticos

- Encapsulamento

- Identifica subárvores potencialmente útil;
- Dá um nome para que possa ser referenciada futuramente.

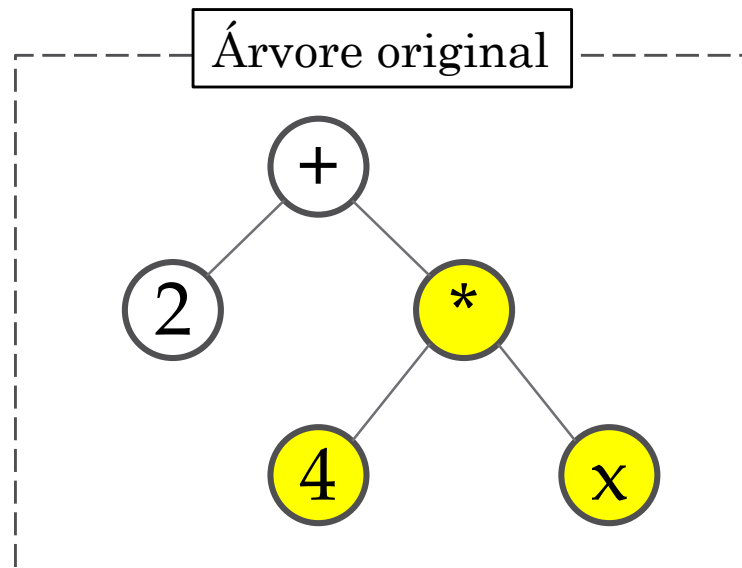




# Operadores genéticos

- Encapsulamento

- Identifica subárvores potencialmente útil;
- Dá um nome para que possa ser referenciada futuramente.

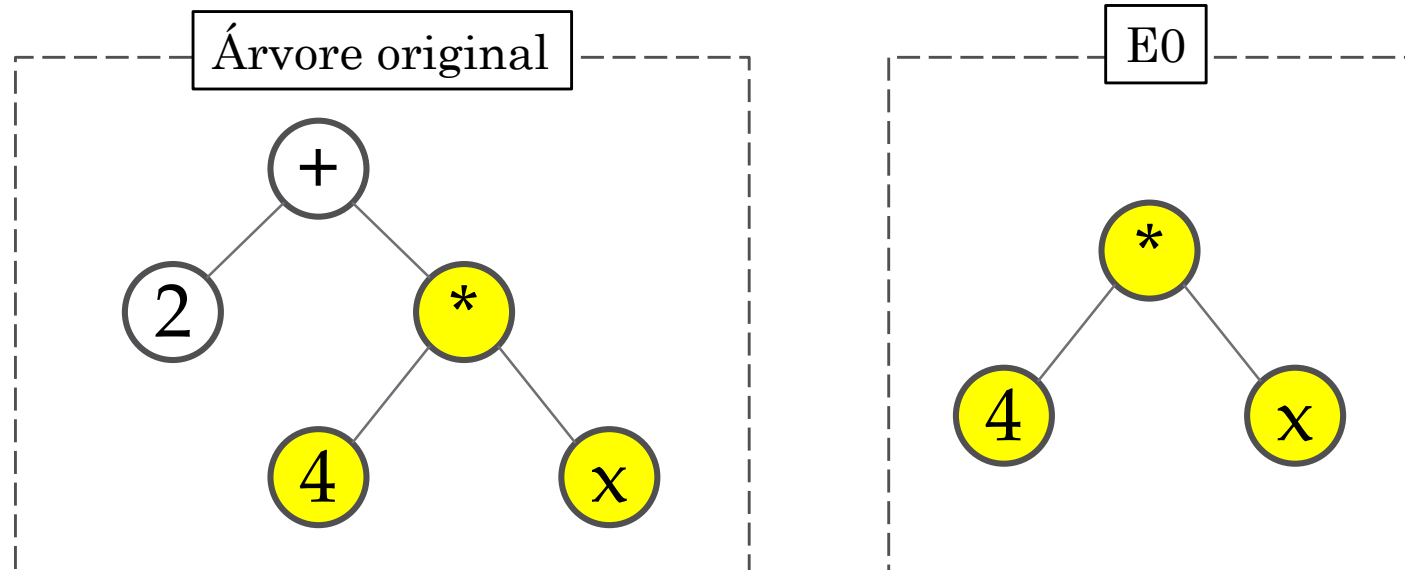




# Operadores genéticos

- Encapsulamento

- Identifica subárvores potencialmente útil;
- Dá um nome para que possa ser referenciada futuramente.



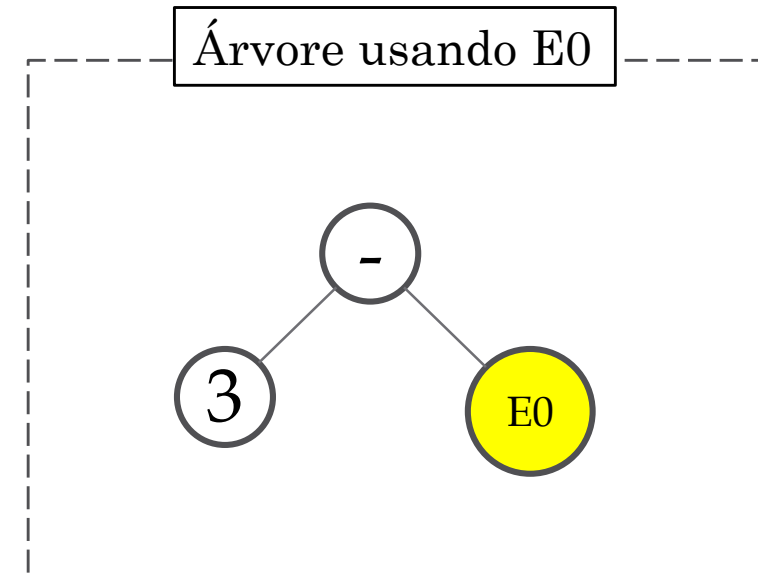
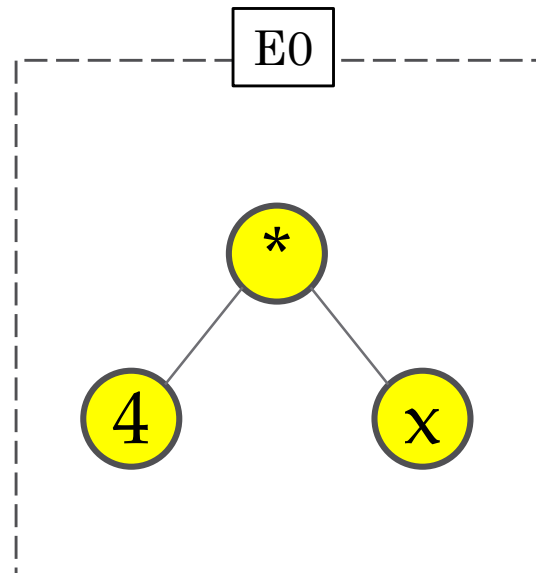
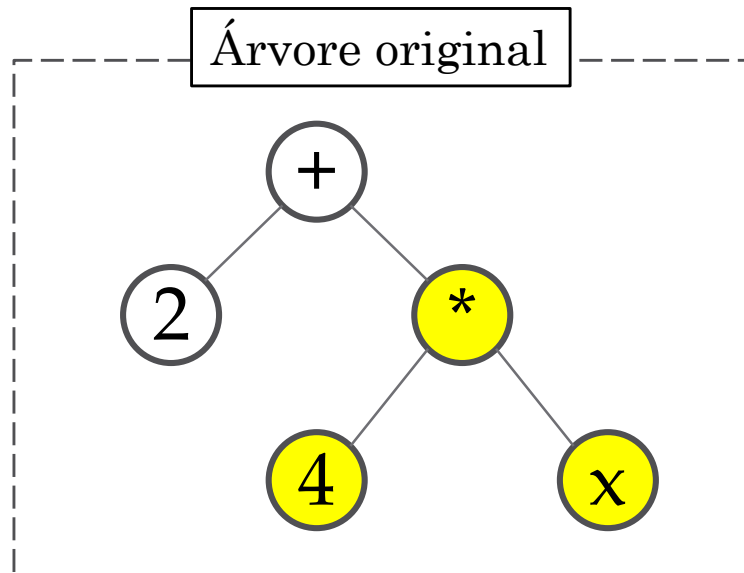




# Operadores genéticos

- Encapsulamento

- Identifica subárvores potencialmente útil;
- Dá um nome para que possa ser referenciada futuramente.





# Operadores genéticos

- Destruição

- Casos complexos, grande parte da população pode ter um *fitness* muito ruim, causando uma perda de diversidade rápida e um custo computacional muito grande;
- Forma de destruir indivíduos medíocres nas gerações iniciais;
- Parâmetros
  - Quantidade de indivíduos mantidos;
  - Condição em que o operador será invocado;
- Indivíduos sobreviventes são escolhidos com base no *fitness*.



# Referências

- ZUBEN, F. V. Representação e Operadores Evolutivos
- ZUBBEN, F. B. Programação Genética
- KOZA, J.R. Genetic Programming: On the Programming of Computers by means of Natural Selection
- NETO, A. G. Programação Genética
- CRUZ, A. J. O. Algoritmos Genéticos
- MEDEIROS, D. Programação Genética