



Universidade Federal Rural de Pernambuco  
Pró-Reitoria de Pesquisa e Pós-Graduação  
**Programa de Pós-Graduação em  
Informática Aplicada**



# Computação Evolutiva

---

AULA 07 – VARIANTES DE ALGORITMOS EVOLUCIONÁRIOS – PARTE 2

# Roteiro

---

Algoritmos Genéticos

Estratégias de Evolução

Programação Evolucionários

Programação Genética

Evolução Diferencial

# Programação Genética

---

O conceito de Programação Genética (PG) é um pouco mais recente do que as outras variantes de AE e introduz algumas ideias diferentes:

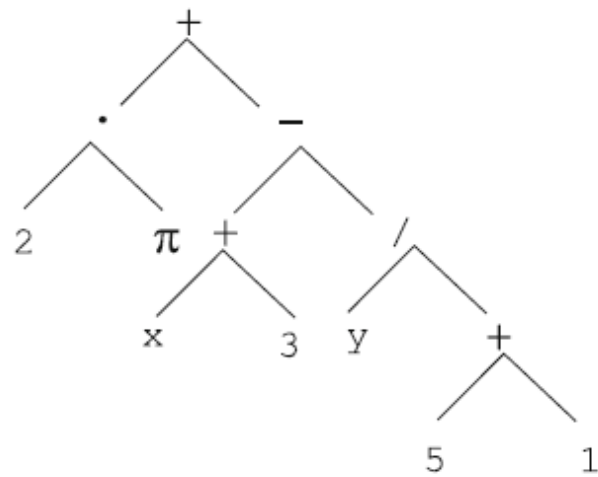
- Os indivíduos são árvores que representam modelos/programas/expressões a serem encontradas
- O *fitness* está relacionado à busca de modelos que melhor representam um conjunto de pontos

Representação	Estrutura em árvore
Recombinação	Troca de subárvores
Mutação	Mudanças aleatórias nas árvores
Seleção dos pais	Proporcional ao fitness
Seleção de sobreviventes	Por geração

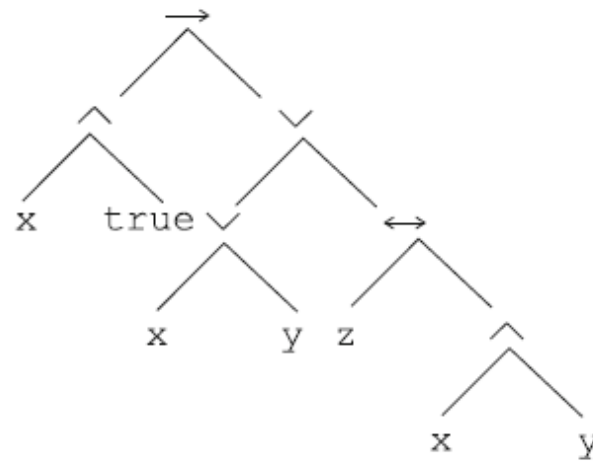
Esqueleto de um algoritmo de PG.

# Programação Genética

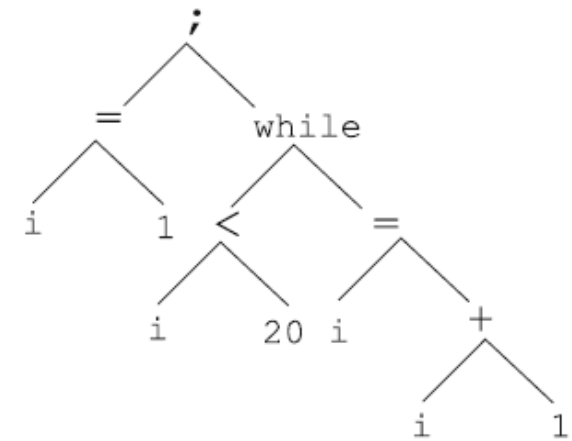
As árvores usadas em PG para representar os cromossomos geralmente estão relacionadas com expressões que possuem alguma sintaxes. Exemplos:



$$2 \cdot \pi + ((x + 3) - \frac{y}{5 + 1})$$



$$(x \wedge true) \rightarrow ((x \vee y) \vee (z \leftrightarrow (x \wedge y)))$$



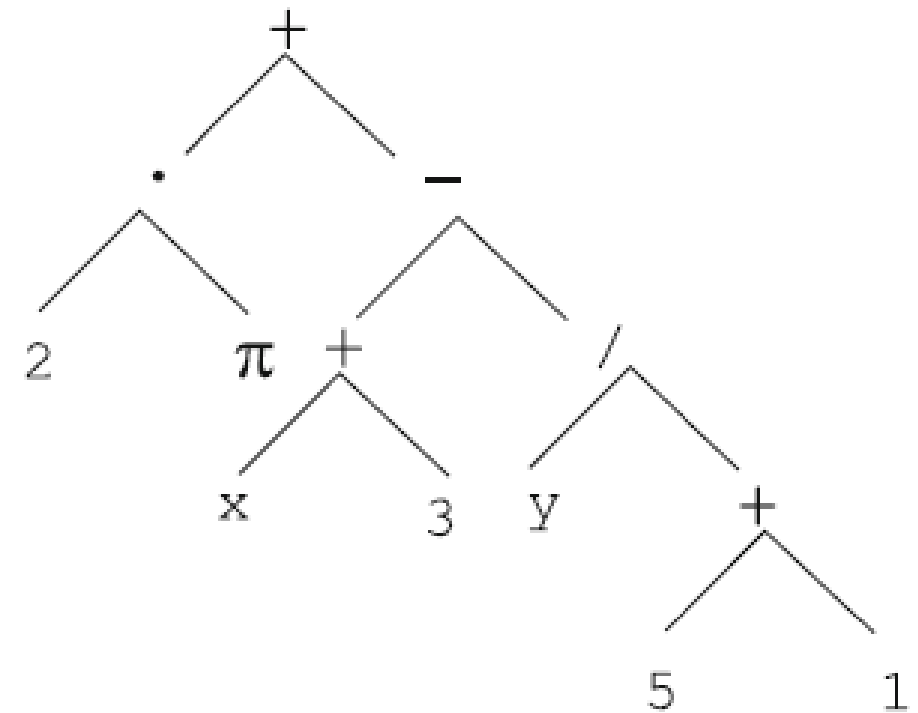
```
i = 1;
while (i < 20)
{
    i = i+1;
}
```

# Programação Genética

A especificação da sintaxe geralmente é feita usando uma gramática, composta por um conjunto de funções e um conjunto de terminais.

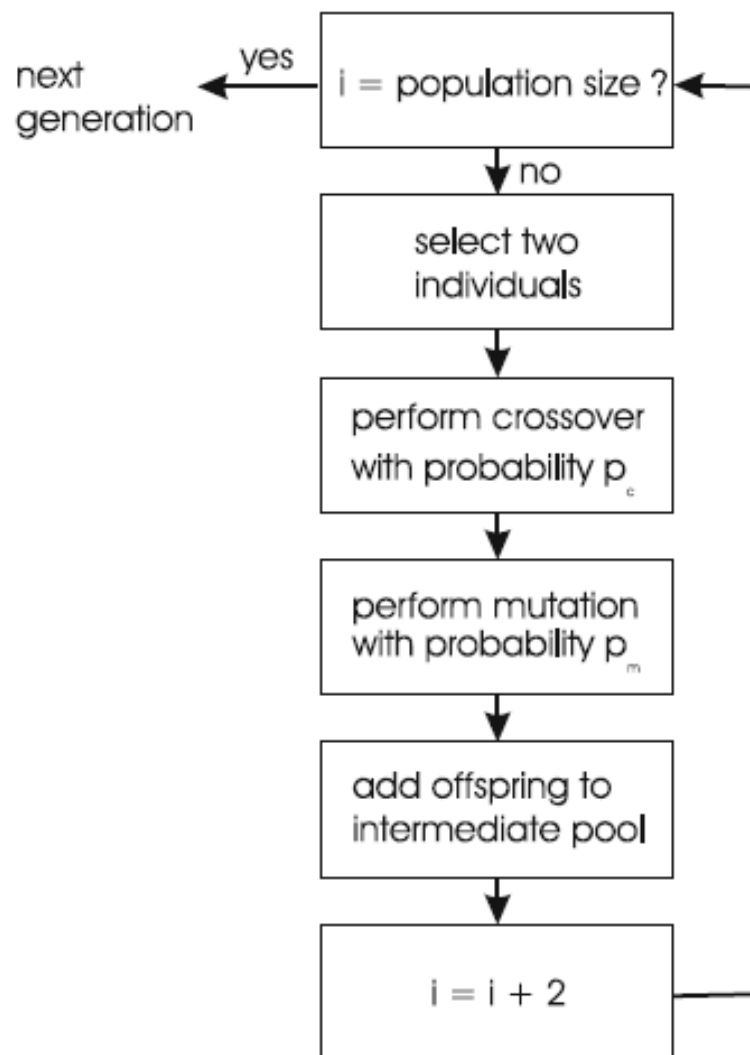
Function set	$\{+, -, \cdot, /\}$
Terminal set	$\mathbb{R} \cup \{x, y\}$

$$2 \cdot \pi + ((x + 3) - \frac{y}{5 + 1})$$

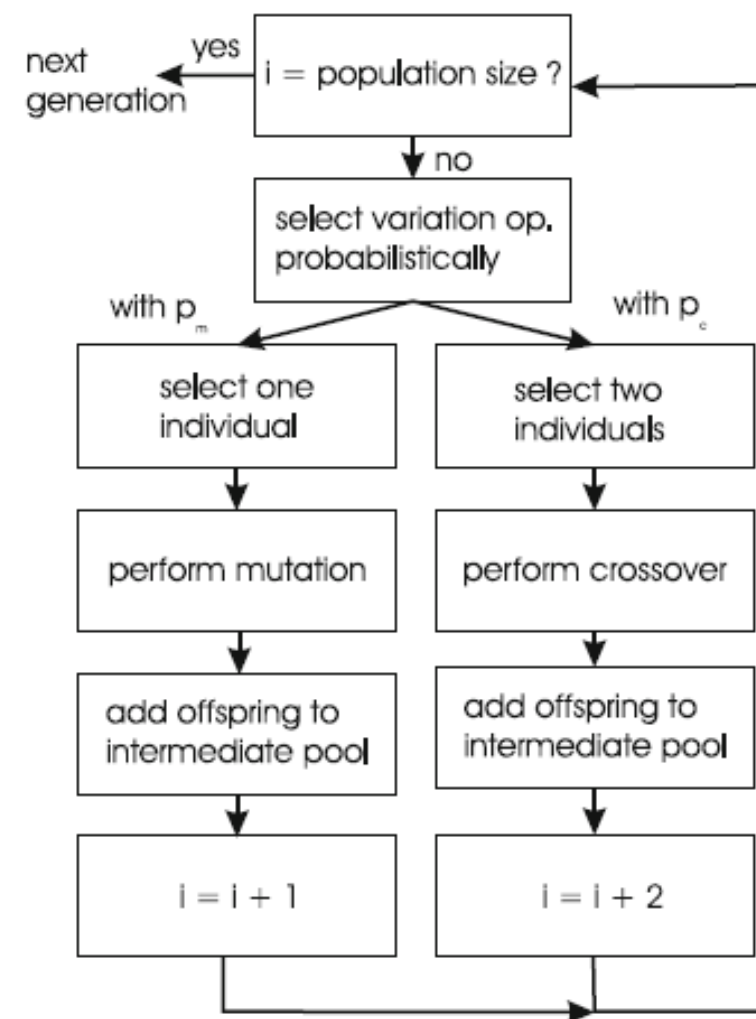


# Programação Genética: fluxo

O método proposto por Koza para variação da população é um pouco diferente do que é feito usando AG:



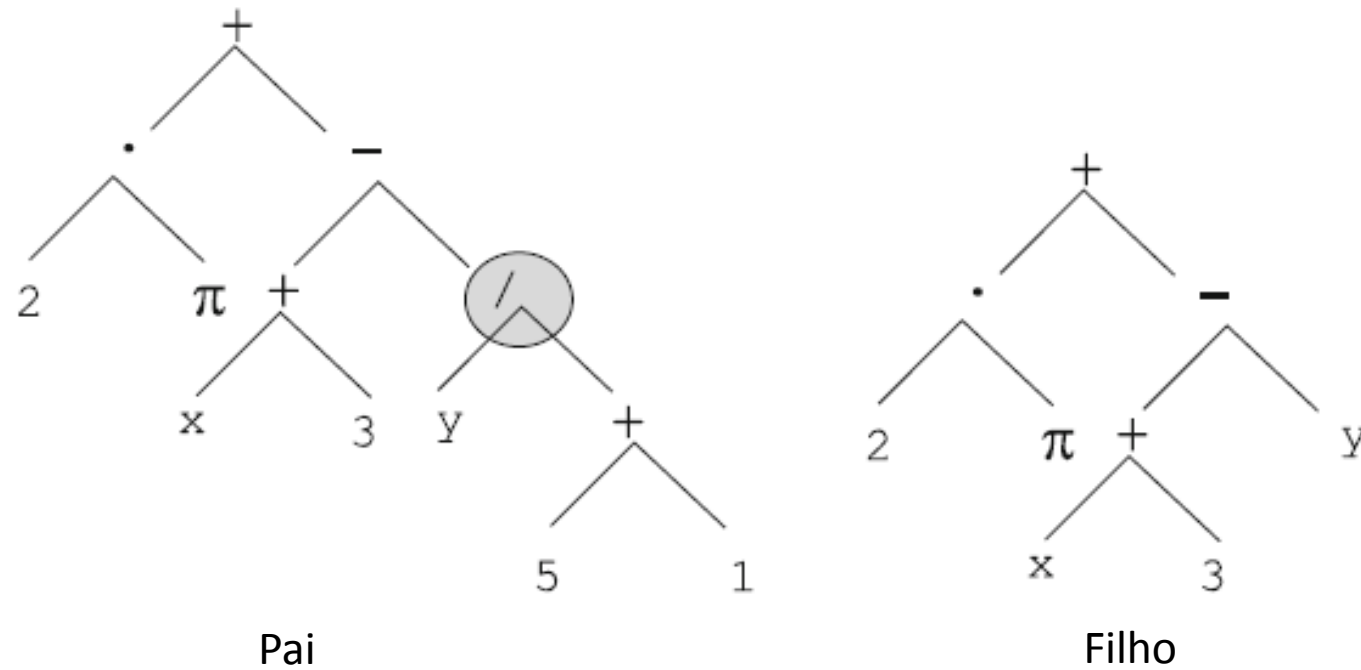
Fluxo padrão do AG



Fluxo padrão para PG

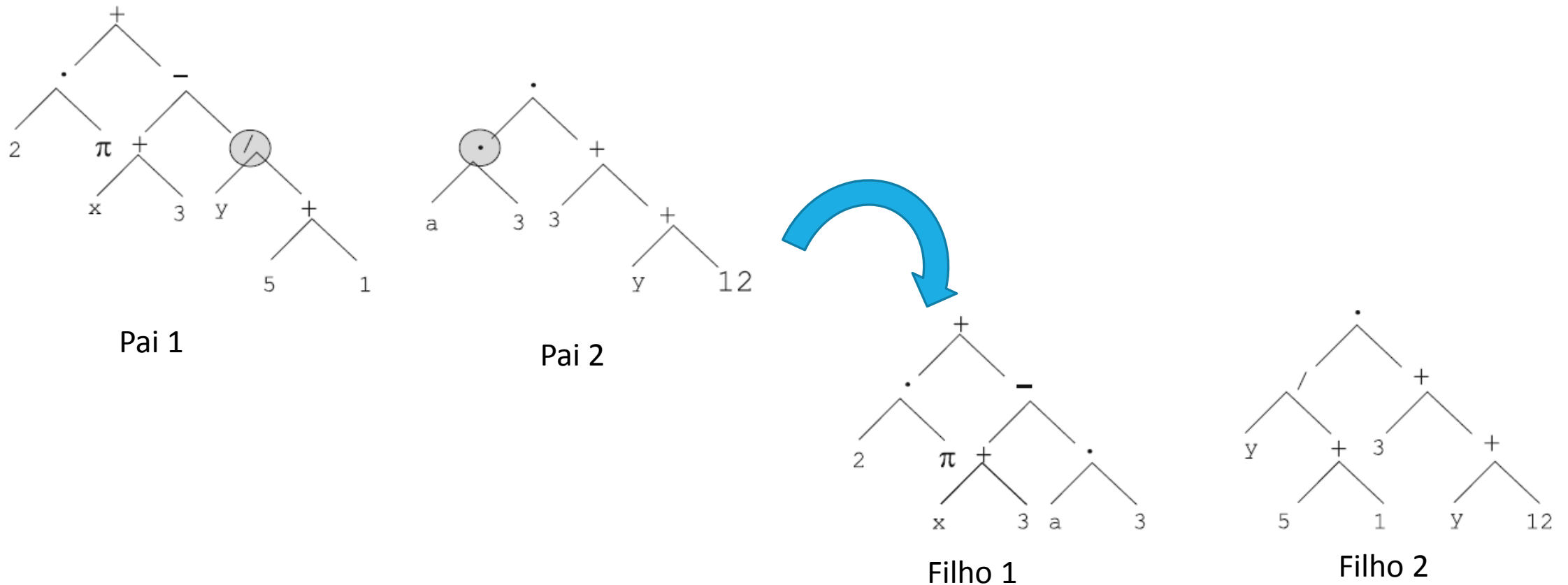
# Programação Genética: mutação

A forma mais comum de realizar mutação em PG consiste em selecionar um nó da árvore aleatoriamente e substituir por uma sub-árvore gerada aleatoriamente.



# Programação Genética: recombinação

A forma mais comum de realizar recombinação é realizando crossover em sub-árvores.





# Programação Genética: *bloat*

---

Este fenômeno é observado em PG quando o tamanho médio das árvores tentem a crescer durante a execução do algoritmo.

- É conhecido como “sobrevivência do mais gordo”
- Existem diversos estudos para entender porque este fenômeno ocorre e tentar contorná-la
  - Pode-se introduzir um fator de penalidade proporcional ao tamanho da árvore gerada após uma mutação ou recombinação

# Evolução diferencial

---

Evolução diferencial (ED) é um algoritmo evolucionário “jovem” que leva no seu nome a lembrança do principal operador usado na metaheurística, que é a mutação diferencial:

- Dada uma população de vetores de soluções candidatas em  $\mathbb{R}^n$ , um novo vetor mutante  $\mathbf{x}'$  é produzido pela adição de um vetor de perturbação:

$$\overline{\mathbf{x}}' = \overline{\mathbf{x}} + \overline{\mathbf{p}},$$

em que o vetor de perturbação é obtido pela multiplicação de um escalar pela diferença de dois outros vetores da população:

$$\overline{\mathbf{p}} = F \cdot (\overline{\mathbf{y}} - \overline{\mathbf{z}}),$$

e o fator de escala  $F > 0$  é um número real que controla a taxa em que a população evolui

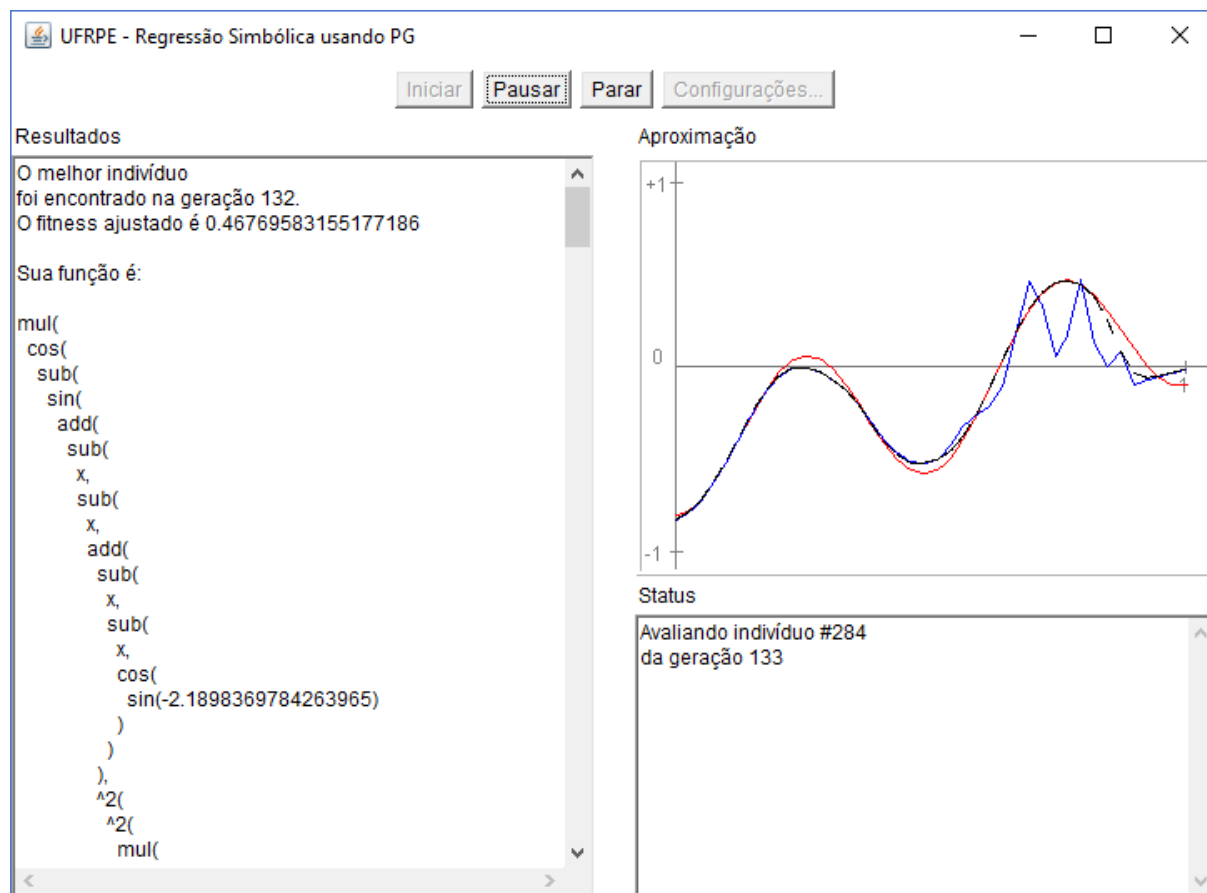
# Evolução diferencial

---

A mutação diferencial é a grande contribuição desta metaheurística e os demais elementos do algoritmos podem ser resumidos da seguinte forma:

Representação	Vetores de valores reais
Recombinação	Cruzamento uniforme
Mutação	Mutação diferencial
Seleção dos pais	Seleção uniforme aleatória dos 3 vetores necessários
Seleção de sobreviventes	Substituição determinística elitista (pais vs. filhos)

Esqueleto de um algoritmo de ED.



# Programação Genética: prática

Koza foi pioneiro no estudo de Programação Genética e existem aplicativos desenvolvidos nos moldes do que foi proposto em seu popular livro sobre o assunto (KOZA, 1992).

- Faça download do projeto compartilhado e execute a classe SRPrincipal
  - Este projeto é uma adaptação do minia aplicativo Java (applet) disponível em <http://alphard.ethz.ch/gerber/approx/default.html>
- A função original predefinida no programa é:

$$f(x) = \sin^2(x) - 0.8 \times \cos^2(2\pi x)$$

# Programação Genética: prática

Escolha uma das seguintes configurações abaixo e executem o algoritmo de regressão simbólica com a função pré-definida no programa:

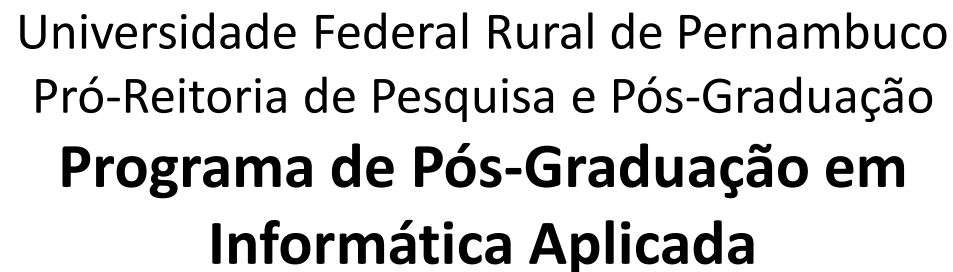
Configuração	Aluno 1	Aluno 2	Aluno 3
Tamanho da população	100	100	Variar
PC	80%	Variar	80%
PM	20%	Variar	20%
Funções	Variar	+, -, x	+, -, x

O objetivo é aproximar a função a seguir:

$$f(x) = \sin^2(x) - 0.8 \times \cos^2(2\pi x)$$



Cada aluno deve apresentar a expressão encontrada (formatada) e o gráfico do aplicativo após o término das execuções!



# Computação Evolutiva

## AULA 07 – VARIANTES DE ALGORITMOS EVOLUCIONÁRIOS – PARTE 2