

1. Computação Evolutiva

Prof. Renato Tinós

Programa de Pós-Graduação Em
Computação Aplicada

Depto. de Computação e Matemática
(FFCLRP/USP)

1.4. Programação Genética

1.4.1. Introdução

1.4.2. Representação por Árvores

1.4.3. Reprodução

1.4.4. Seleção

1.4.5. Exemplos

1.4.1. Introdução

- Programação Genética (*Genetic Programming* – GP)
 - Proposta nos EUA por J. Koza
 - na década de 1990
 - inspirada em AGs
- Aplicada tipicamente em:
 - Tarefas de aprendizado de máquina
- Característica:
 - Uso de cromossomos não-lineares
 - Grafos
 - Árvores
 - Necessita populações muito grandes
 - lenta

1.4.1. Introdução

Sumário: características geralmente encontradas

Representação	Árvores
Recombinação	Troca de sub-árvores
Mutação	Mudanças aleatórias nas árvores
Seleção dos pais	Proporcional ao fitness
Seleção para próxima população	Geracional

1.4.1. Introdução

Exemplo 1.4.1.

- Um banco quer distinguir bons e maus investidores
- Modelo será baseado nos dados já conhecidos

ID	Número de filhos (NF)	Salário (S)	Estado civil (EC)	OK?
ID-1	2	45000	Casado	0
ID-2	0	30000	Solteiro	1
ID-3	1	40000	Divorciado	1
...				

1.4.1. Introdução

Exemplo 1.4.1.

- Um possível modelo:

```
IF (NF = 2) AND (S > 80000) THEN OK=1 ELSE  
OK=0
```

- Em geral:

```
IF formula THEN OK=1 ELSE OK=0
```

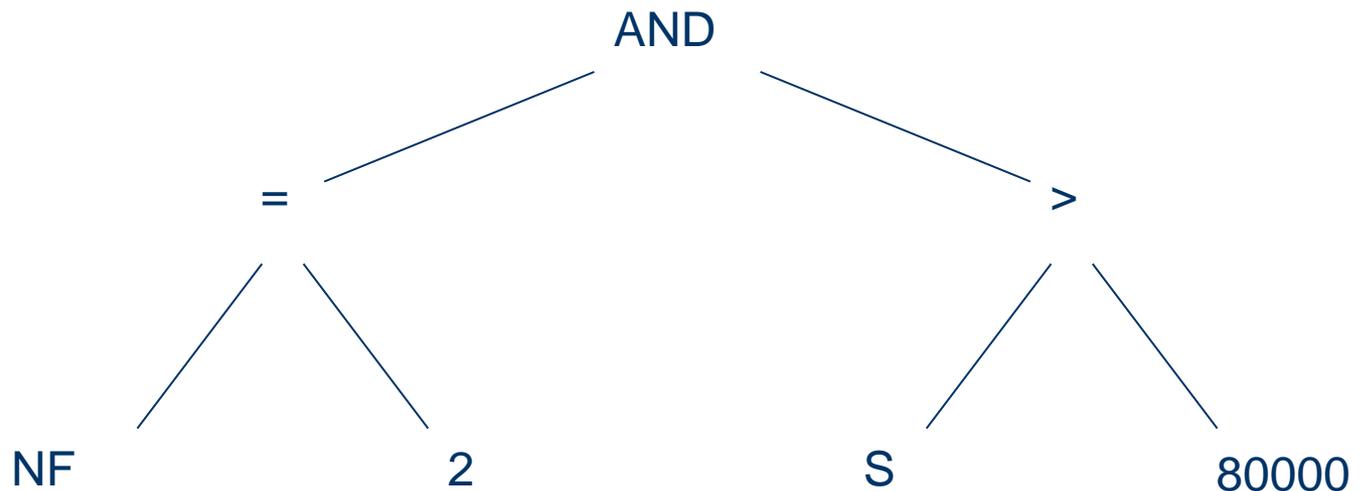
- O que é desconhecido é a fórmula
 - O espaço de busca (fenótipos) é portanto o conjunto de fórmulas candidatas
 - Fitness de uma fórmula:
 - Porcentagem dos acertos de classificação dados pela fórmula sobre o conjunto de treinamento
 - Representação natural das fórmulas (genótipos): árvores

1.4.1. Introdução

Exemplo 1.4.1.

IF (NF = 2) AND (S > 80000) THEN OK=1 ELSE OK=0

Pode ser representada pela seguinte árvore:



1.4.2. Representação por Árvores

- Árvores podem representar diferentes formas de representação do conhecimento

- Fórmula aritmética

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

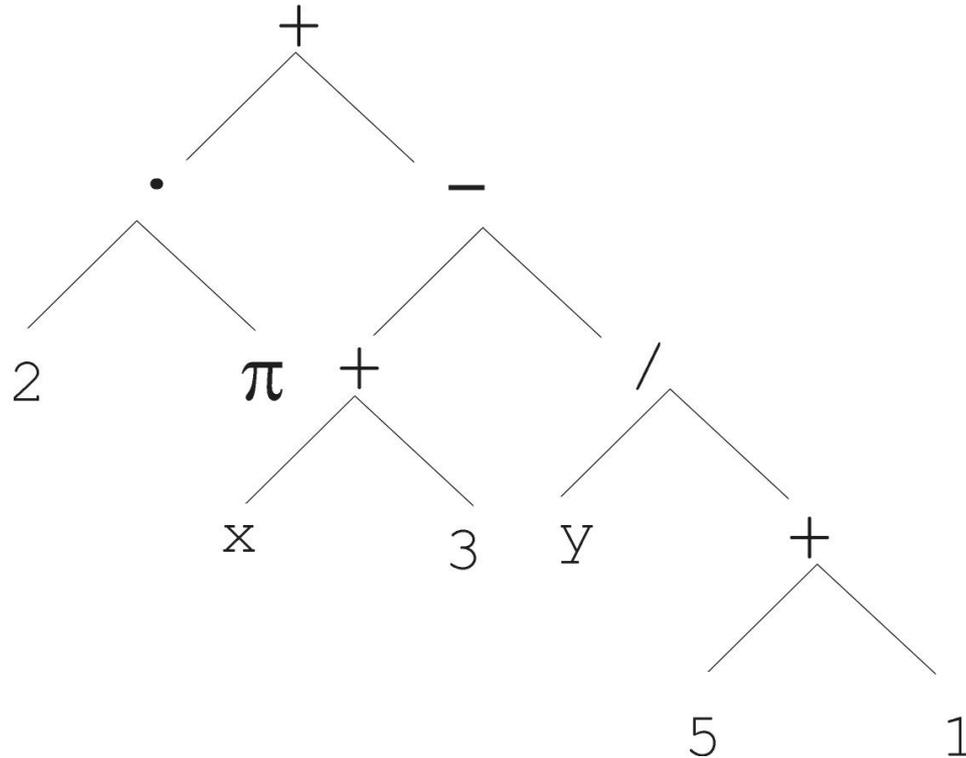
- Fórmula lógica

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

- Programa

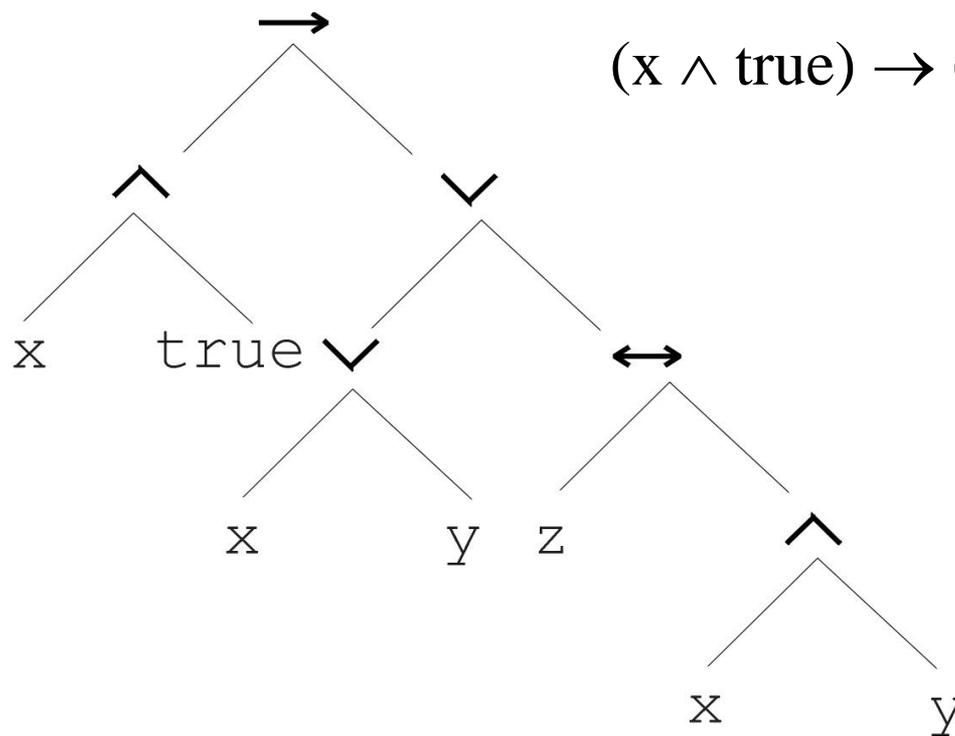
```
i = 1;
while (i < 20)
{
    i = i + 1
}
```

1.4.2. Representação por Árvores

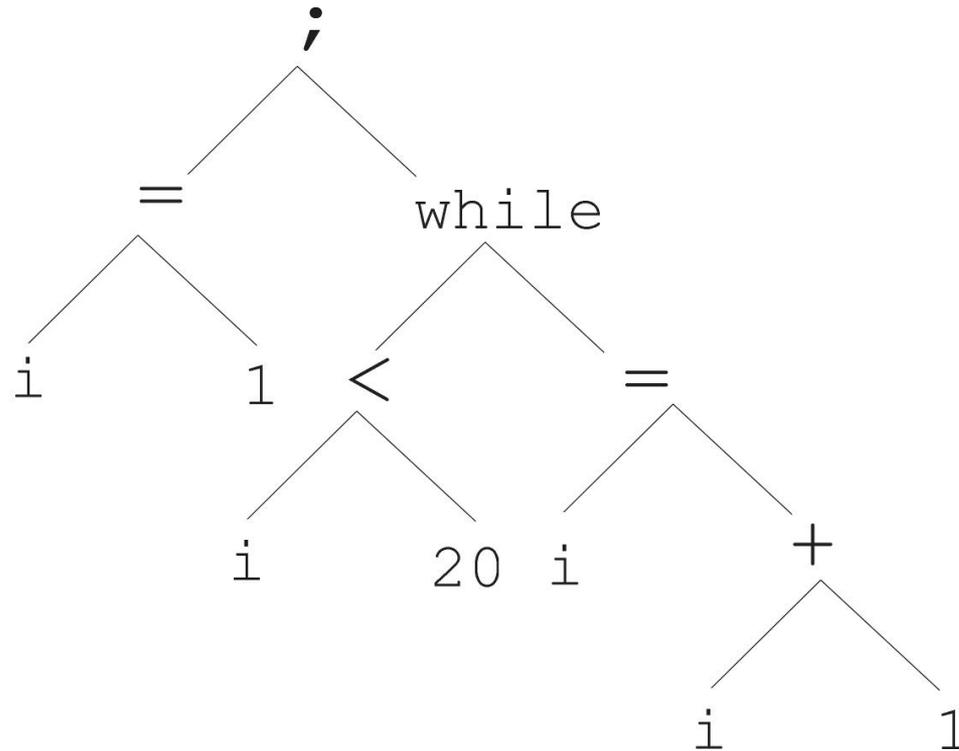


$$2 \cdot \pi + \left((x + 3) - \frac{y}{5 + 1} \right)$$

1.4.2. Representação por Árvores



1.4.2. Representação por Árvores



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

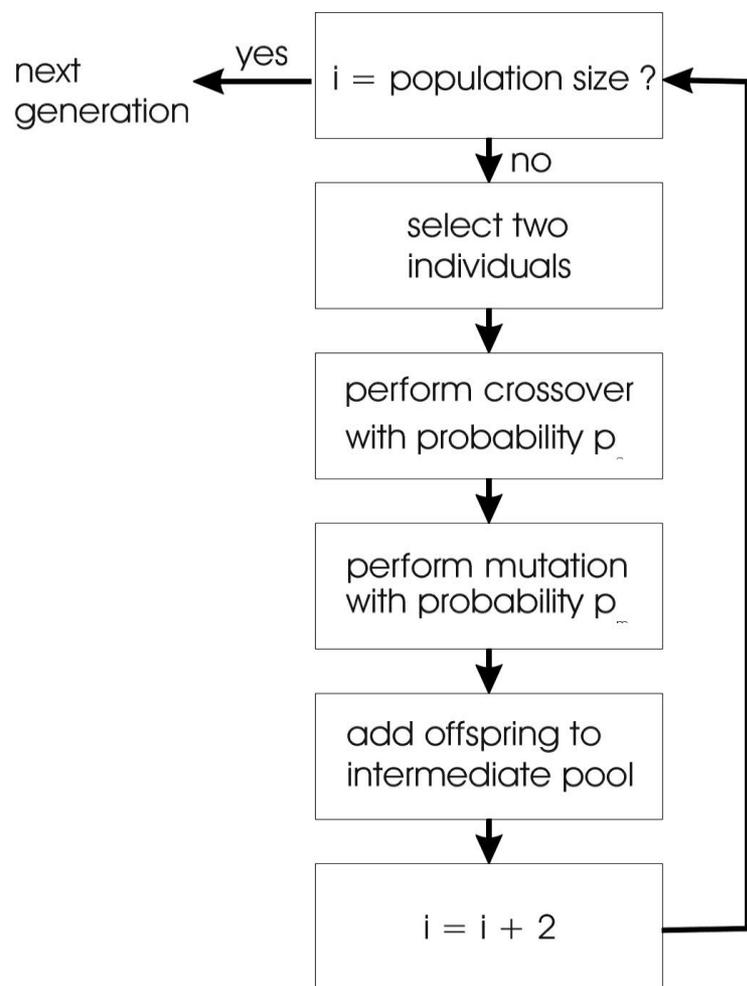
1.4.2. Representação por Árvores

- Em AGs, ESs e EP, os cromossomos são representados por estruturas lineares
 - Em AGs, ESs e EP os tamanhos dos cromossomos são geralmente fixos
- Cromossomos em forma de árvore são estruturas não-lineares
 - Árvores em GP podem variar em profundidade e largura
- Definição dos nós
 - Nós intermediários: conjunto F
 - Exemplo: $F = \{ + , - , \text{sqrt} , / \}$
 - Nós terminais: conjunto T
 - Exemplo: $T = R$

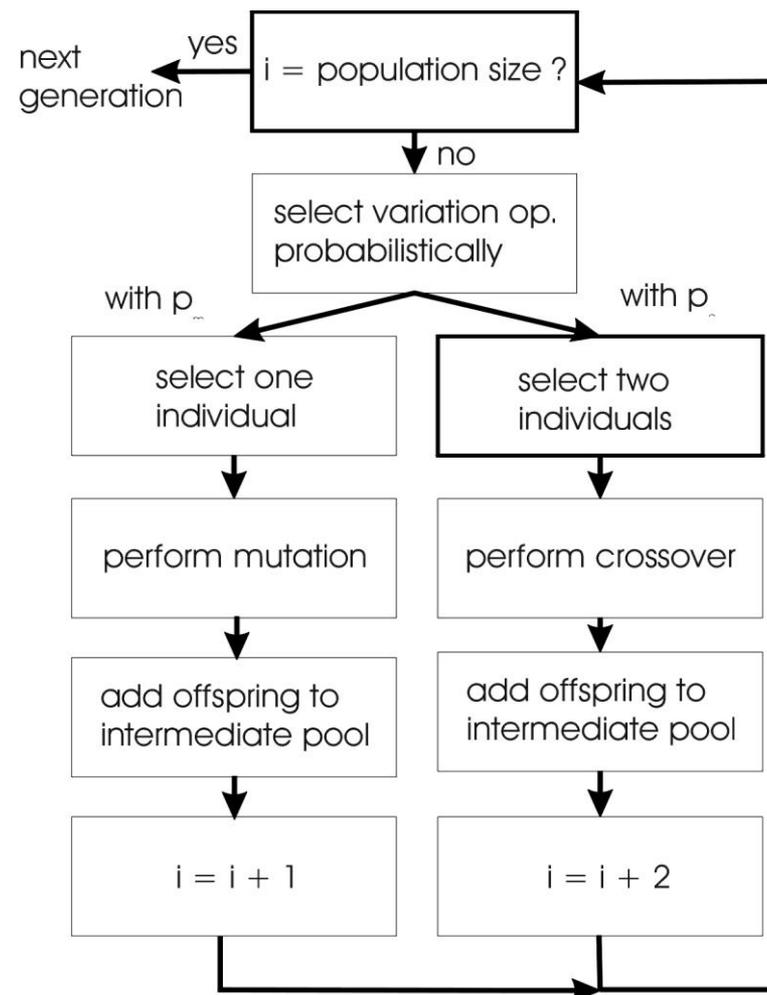
1.4.3. Reprodução

- AG usa crossover E mutação sequencialmente
 - Ocorrem com certa probabilidade
- GP usa crossover OU mutação
 - Escolhidos com certa probabilidade

1.4.3. Reprodução



AG

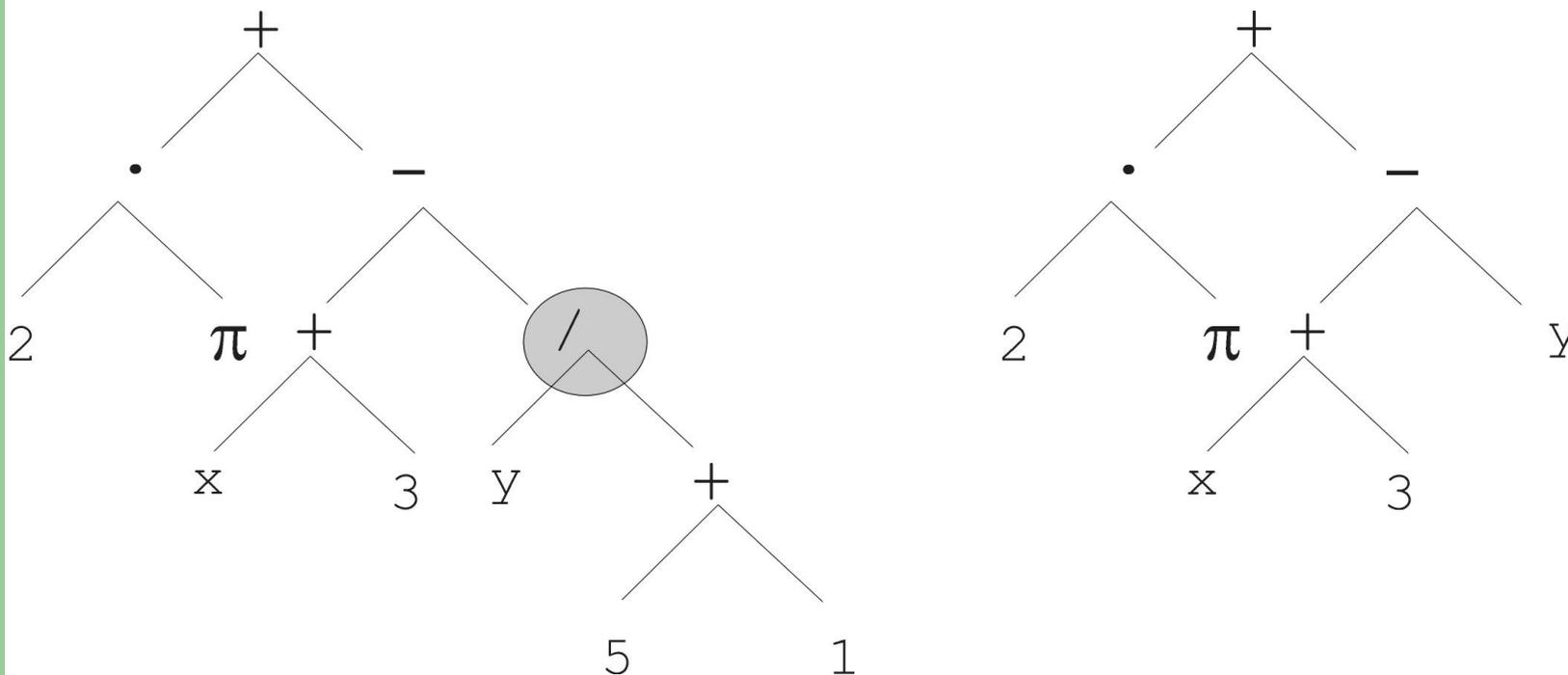


GP

1.4.3. Reprodução

- Mutaç o

- Forma mais comum: substitua aleatoriamente uma sub rvore por uma  rvore gerada aleatoriamente



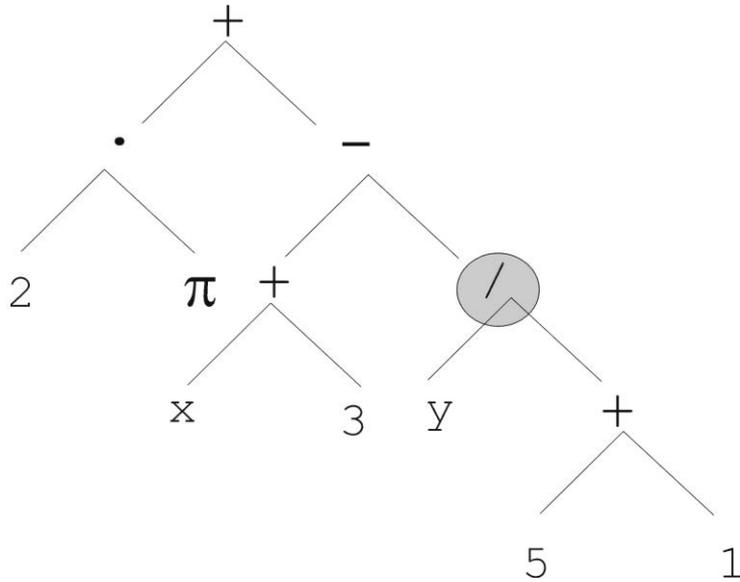
1.4.3. Reprodução

- Mutação
 - Tem dois parâmetros
 - Probabilidade p_m de se escolher mutação ao invés de recombinação
 - p_m é recomendada como 0 (Koza'92) ou muito próxima disso, e.g. 0,05 (Banzhaf et al. '98)
 - Probabilidade de escolher um ponto interno como raiz da subárvore a ser substituída
 - Observação: o tamanho de um filho pode exceder o tamanho do pai

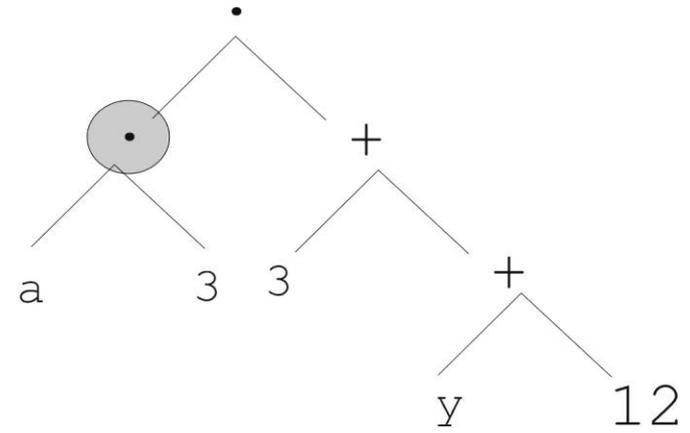
1.4.3. Reprodução

- Recombinação

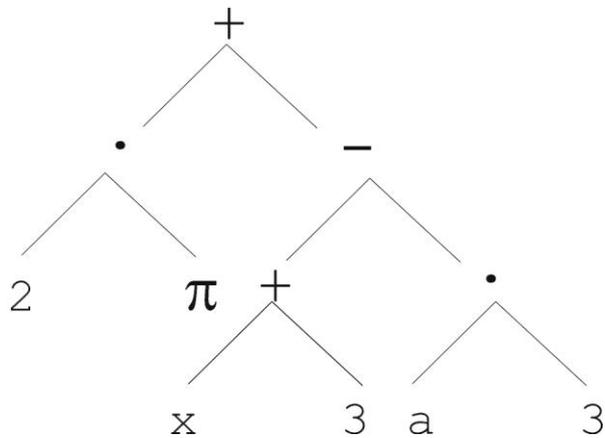
- Forma mais comum: troca de duas subárvores dos dois pais escolhidas aleatoriamente
- Tem dois parâmetros
 - Probabilidade p_c de escolher recombinação ao invés de mutação
 - Probabilidade de escolher um ponto interno como ponto de crossover
- Observação: o tamanho de um filho pode exceder o tamanho do pai



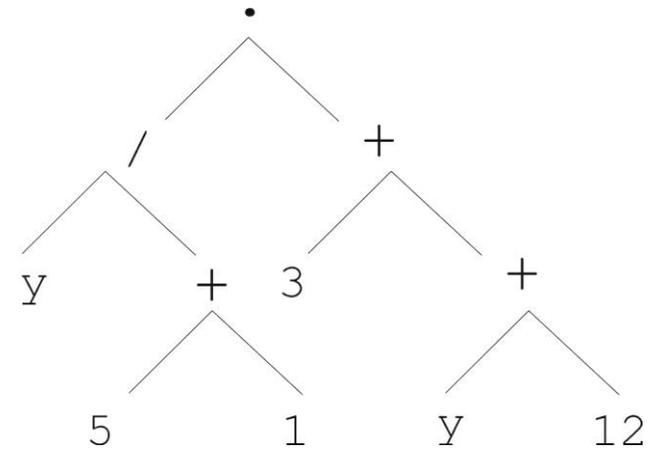
Pai 1



Pai 2



Filho 1



Filho 2

1.4.4. Seleção

- Seleção dos pais a serem reproduzidos
 - Tipicamente é feita proporcionalmente ao fitness dos pais
 - Sobre-seleção (*Over-selection*) em populações muito grandes
 - Ranqueie população de acordo com o fitness e a divida em dois grupos
 - Grupo 1: melhores $x\%$ da população
 - Para tamanhos da população = 1000, 2000, 4000, 8000 : $x = 32\%, 16\%, 8\%, 4\%$
 - Grupo 2: $(100-x)\%$ indivíduos restantes
 - 80% das operações de seleção escolhidas no Grupo 1
 - 20% no Grupo 2

1.4.4. Seleção

- Seleção para a próxima geração
 - Esquema geracional
 - Como no AG
 - As vezes o esquema estacionário é usado
 - Permite o elitismo

1.4.5. Exemplos

Exemplo 1.4.2. (Eiben & Smith, 2003) Regressão simbólica

- Dados n pontos no \mathbf{R}^2 , $(x_1, y_1), \dots, (x_n, y_n)$
- Acha função $f(x)$ sujeito a $\forall i = 1, \dots, n : f(x_i) = y_i$
- Possível solução usando GP:

- Representação

- ❖ $F = \{+, -, /, \sin, \cos\}$,

- ❖ $T = \mathbf{R} \cup \{x\}$

- Fitness dado pelo erro

$$err(f) = \sum_{i=1}^n (f(x_i) - y_i)^2$$

- Uso de operadores padrão

- Tamanho da população = 1000

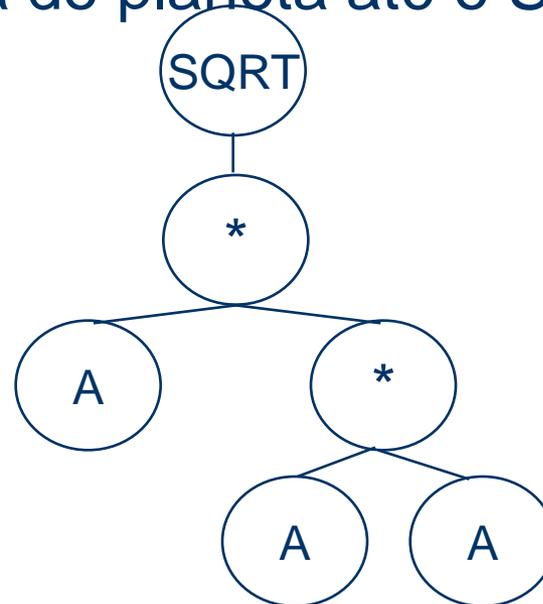
- Critério de parada

- n “hits” ou 50000 avaliações de fitness

- “hit” é quando $|f(x_i) - y_i| < 0.0001$

1.4.5. Exemplos

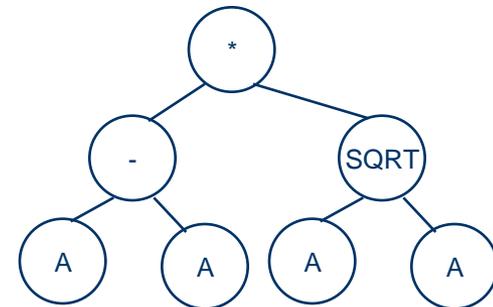
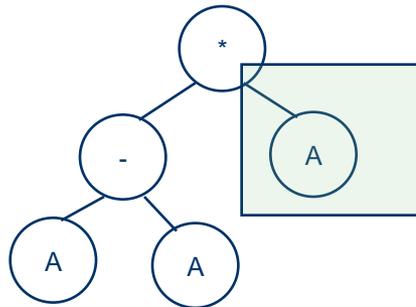
Exemplo 1.4.3. (MITCHELL, 1996) Imagine que desejamos chegar na expressão, em LISP, para o período orbital de um planeta no sistema solar ($\text{SQRT}(*A(*AA))$), sendo A a distância média normalizada do planeta até o Sol



1.4.5. Exemplos

Exemplo 1.4.3.

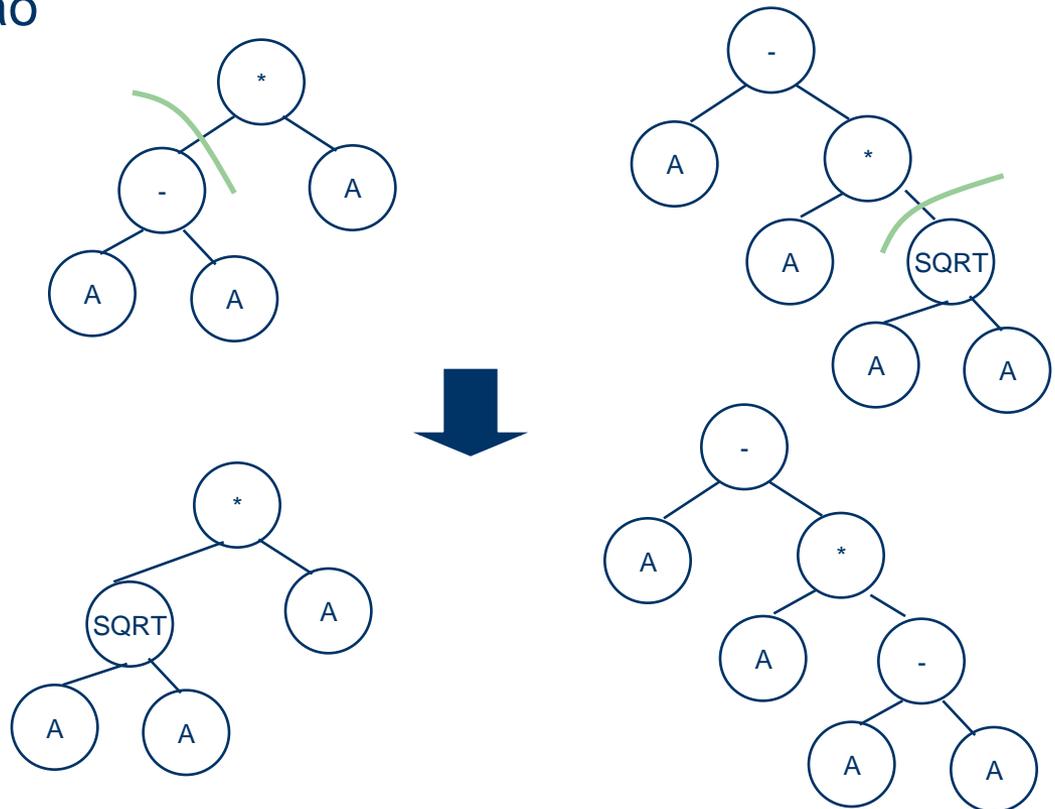
- $F = \{ \text{SQRT}, +, -, *, / \}$
- $T = \{A\}$
- População inicial com várias árvores aleatórias
- Fitness: semelhante ao problema anterior
- Mutação padrão
 - Exemplo:



1.4.5. Exemplos

Exemplo 1.4.3.

- Crossover padrão



1.4.5. Exemplos

- Discussão
 - GP é:
 - A arte de evoluir programas de computadores ?
 - Meio de automatizar a programação de computadores?
 - AG com outra representação?

Comentários

- Referências
 - Eiben, A. E. & Smith, J. E. *Introduction to Evolutionary Computation*. Springer, 2003
 - Capítulo 6
 - Mitchell, M. *An introduction to genetic algorithms*. MIT Press, 1996
 - Capítulo 2, Seção 2.1
- Agradecimentos
 - Parte do material desta apresentação foi obtida através de
 - Material de apoio do livro Eiben, A. E. & Smith, J. E. *Introduction to Evolutionary Computation*. Springer, 2003.